

UNIVERSITE DE MONTPELLIER

FACULTE DES SCIENCES

T H E S E

présentée à la Faculté des Sciences de Montpellier
pour obtenir le grade de Docteur ès Sciences Appliquées
(INFORMATIQUE)

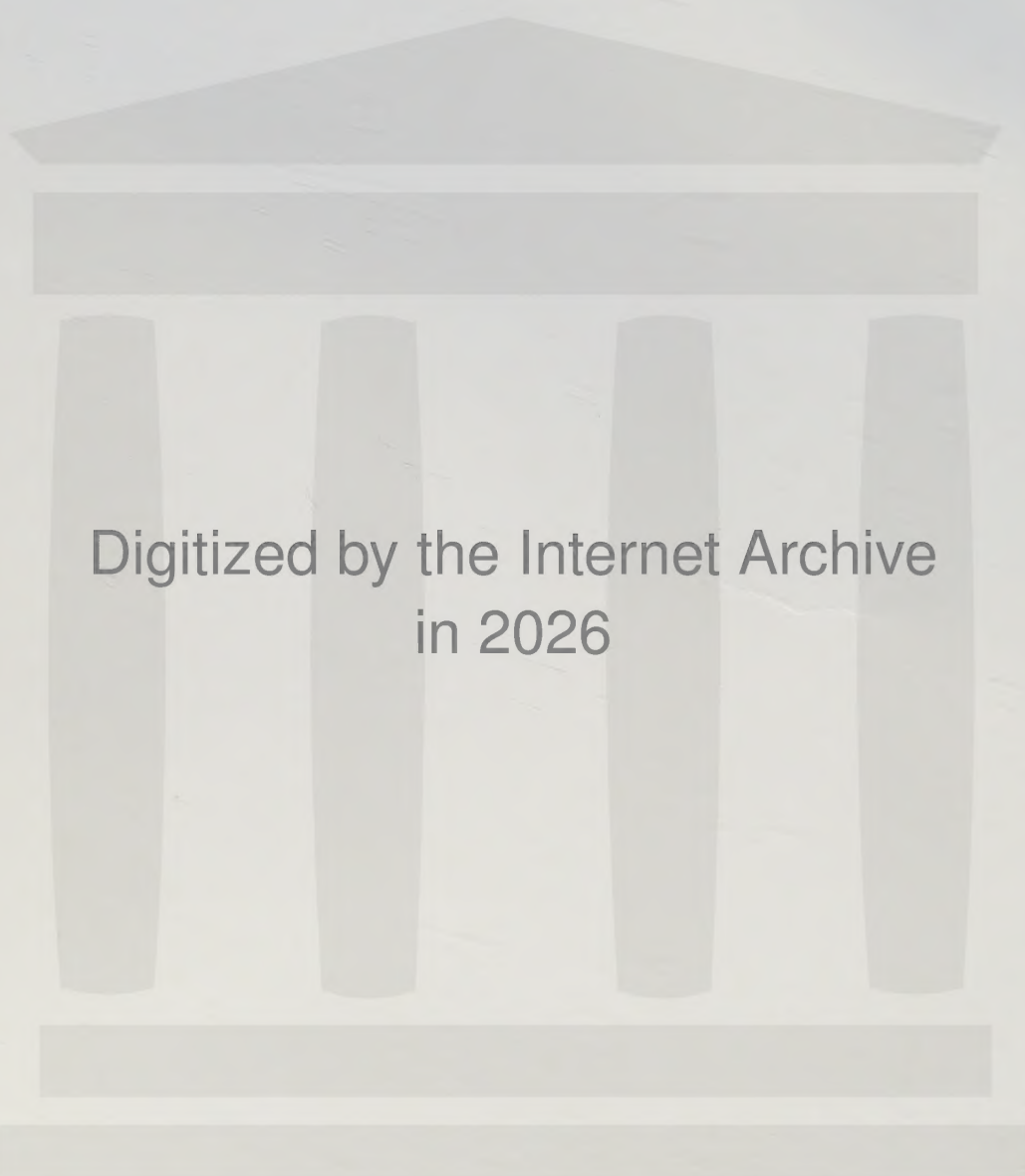
CONCEPTION ET REALISATION
D'UN SYSTEME DE DOCUMENTATION AUTOMATIQUE
POUR LA JURISPRUDENCE

par

Bernard FILLIATRE

JURY : MM. B. CHARLES,	Professeur	Président
J. ARSAC,	Professeur	
J. FALGUEIRETTES,	Professeur	} Assesseurs
F. LAPSCHER,	Maître de Conférences	
P. CATALA,	Professeur	Invité

— JUIN 1970 —



Digitized by the Internet Archive
in 2026

https://archive.org/details/IA41548307_0017

UNIVERSITE DE MONTPELLIER

FACULTE DES SCIENCES

T H E S E

présentée à la Faculté des Sciences de Montpellier
pour obtenir le grade de Docteur ès Sciences Appliquées
(INFORMATIQUE)

CONCEPTION ET REALISATION
D'UN SYSTEME DE DOCUMENTATION AUTOMATIQUE
POUR LA JURISPRUDENCE

par

Bernard FILLIATRE

JURY : MM. B. CHARLES,	Professeur	Président
J. ARSAC,	Professeur	
J. FALGUEIRETTES,	Professeur	} Assesseurs
F. LAPSCHER,	Maître de Conférences	
P. CATALA,	Professeur	Invité

— JUIN 1970 —

UNIVERSITEIT DE WOLFFELITE

WOLFFELITE DE WOLFFELITE

WOLFFELITE

WOLFFELITE DE WOLFFELITE
WOLFFELITE DE WOLFFELITE
WOLFFELITE DE WOLFFELITE

UNIVERSITEIT DE WOLFFELITE

WOLFFELITE DE WOLFFELITE

WOLFFELITE DE WOLFFELITE



Doyen : P. DUMONTET

Assesseurs : Ch. SAUVAGE - M. MAURIN

Doyen honoraire : P. MATHIAS

Professeurs honoraires :

R. JACQUES	J. SOULAS	G. CAUQUIL
M. CASTERAS	G. DENIZOT	
E. CARRIERE	J. GRANIER	
E. TURRIERE	Ch. BOUHET	

Secrétaire Général : E. SIAU

Professeurs titulaires :

M. M. MOUSSERON	Chimie Organique
M. J.P. ROIG	Physique
M. P. CHATELAIN	Minéralogie & Cristallog.
Mle O. TUZET	Zoologie
M. G. COUCHET	Mécanique Supérieure
M. J. SALVINIEN	Chimie-Physique
Mle A.M. VERGNOUX	Physique
M. J. AVIAS	Géologie
M. P. VIELES	Chimie
M. J. RABINOVITCH	Physique (C.S.U.P.)
M. E. KAHANE	Chimie Biologique
M. J.J. MOREAU	Mécanique rationnelle
M. B. CHARLES	Mathématiques Pures
M. J. MOTTE	Biologie Végétale
M. R. JOUTY	Physique
M. P. DUMONTET	Physique
M. R. LEGENDRE	Zoologie
M. I. ASSENMACHER	Physiologie Animale
M. A. CASADEVALL	Chimie Organique
M. B. PISTOULET	Physique
M. Ch. ROUMIEU	Analyse Supérieure
M. J. ROBIN	Physique
M. A. POTIER	Labor. des Acides Minéraux

M. R. LAFONT	Physique
M. R. JACQUIER	Chimie
M. J. FALGUEIRETTES	Minéralogie-Cristallographie
M. J. REGNIER	Chimie
Mme J. CHARLES	Mathématiques
M. P. CAILLON	Physique
M. J. ROUZAUD	Chimie
M. Ch. SAUVAGE	Botanique
M. H. CHRISTOL	Chimie (E.N.S.I)
Mme G. VERNET	Biologie Animale
M. L. CECCHI	Physique
M. H. ANDRILLAT	Astronomie
M. M. SAVELLI	Physique
M. M. MATTAUER	Géologie
M. L. EUZET	Zoologie
M. C. DELOUPY	Physique
M. L. GRAMBAST	Botanique
M. A. BONNET	Botanique
M. G. LAMATY	Chimie
M. R. MARTY	Psychophysiologie

Professeurs sans chaire :

Mme M.L. DE POUQUES	Botanique (C.S.U.P.)
M. P. MURAOUR	Géophysique Appliquée
M. G. TOURNE	Chimie
M. J. REMY	Géologie
M. P. DEMANGEON	Géologie
Mme S. ROBIN	Physique
Mme Ch. DUBOUL-RAZAVET	Géologie (C.S.U.P.)
M. E. GROUBERT	Physique
Mme H. GUASTALLA	Biologie Physico-Chimique
M. F. PROUST	Géologie
Mle A. BAILLETTE	Mathématiques (C.S.U.P.)
M. A. BASSOMPIERRE	Physique
M. M. MAURIN	Chimie
M. N. ROBY	Mathématiques

Mme N. PARIS	Physiologie Végétale
M. J. PARIS	Zoologie
M. R. GAUFRES	Chimie
M. J. SOULIER	Chimie (C.S.U.P.)
M. G. BOUGNOT	Physique
M. R. LENEL	Biologie Animale

Professeurs Associés :

M. E. AKUTOWICZ	Mathématiques
M. A. MICALI	Mathématiques
M. C. VAGO	Biologie Animale
M. K. KOHLER	Biochimie

Maitres de Conférences :

M. P. MOLINO	Mathématiques
M. Y. GUITTON	Physiologie Végétale (CSUP)
M. G. BOMBRE	Physique (C.S.U.P.)
M. C. PUISSEGUR	Biologie Animale (C.S.U.P.)
M. P. VIALLET	Chimie (C.S.U.P.)
M. J.M. MORETTI	Chimie
M. L. THALER	Géologie
M. J. LEGRAND	Physiologie Animale
M. M. ROUZEYRE	Physique
M. P. SABATIER	Mathématiques
M. Ch. CASTAING	Mathématiques
M. J.D. BAYLE	Physiologie Animale
Mme M. HAKIM	Mathématiques
M. R. HAKIM	Mathématiques
M. G. BAUSSET	Mathématiques (C.S.U.P.)
M. G. LOUPIAS	Mathématiques
M. G. MASCHERPA	Chimie
M. C. GOUT	Physique
M. R. JONARD	Botanique
M. F. LAPSCHER	Physique
M. C. COMBES	Zoologie (C.S.U.P.)
M. B. BRUN	Chimie Physique
M. L. KAISER	Physique (C.S.U.P.)

M. M. DENIZOT	Biologie Végétale
M. L. GIRAL	Chimie
M. A. COMMEYRAS	Chimie
M. M. AMANIEU	Sciences et Techniques de l'Eau (Hydrobiologie et Mariculture).

Maître de Conférences associé :

M. Y. CORMARY	Géologie
---------------------	----------

Chargés d'Enseignement :

M. M. LEFRANC	Mathématiques
M. R. MARTY	Mathématiques
M. PIETRASSANTA	Chimie

INSTITUT UNIVERSITAIRE DE TECHNOLOGIE

Professeurs titulaires :

M. R. CORRIU	Chimie
--------------------	--------

Maîtres de Conférences :

M. R. CANO	Physique
M. FILLARD	Physique
M. GARCIA	Métallurgie-Génie Mécaniq.
M. J.P. TRILLES	Biologie
M. G. BORDURE	Physique
M. L. LASSABATERE	Electricité-Electronique
M. G. LECOY	Génie Electrique

Maître de Conférences associé :

M. JOUANNA	Génie Civil
------------------	-------------

Chargé d'Enseignement :

M. Y. ESCOUFIER	Mathématiques
-----------------------	---------------

Ce travail a été effectué au laboratoire de Calcul de la Faculté des Sciences de Montpellier en liaison avec l'Institut de recherche et d'Etudes pour le Traitement de l'Information Juridique, de la Faculté de Droit de Montpellier.

La partie Informatique a été faite sous la Direction de Monsieur le Professeur FALGUEIRETTES auquel je tiens à exprimer toute ma reconnaissance pour la formation qu'il m'a permis d'acquérir et les nombreux conseils dont il m'a fait bénéficier. Je me plais à souligner ici le dynamisme dont il a fait preuve dans la création et le développement du laboratoire de Calcul.

La partie Juridique fut dirigée avec une grande compétence par M. le Professeur CATALA que je tiens à remercier tout particulièrement pour son aide efficace et pour l'élargissement des horizons qu'il m'a procuré par ce travail.

Que Monsieur le Professeur CHARLES qui a bien voulu assumer la présidence de ce Jury trouve ici l'expression de ma profonde gratitude.

Je tiens à remercier Monsieur le Professeur ARSAC qui, en dépit de ses occupations, m'a fait le grand honneur de se pencher sur ce travail, de m'éclairer de ses conseils et d'accepter de venir juger le résultat de mes recherches.

Ma reconnaissance va aussi à Monsieur le Professeur LAPCHER pour l'aide qu'il a bien voulu m'apporter lors de la rédaction de cette thèse.

Que tous les chercheurs et techniciens , tant du laboratoire de Calcul que de l'IRETIIJ, trouvent ici l'expression de mes remerciements pour le climat amical et l'esprit d'équipe qui a toujours régné entre nous. Ne pouvant pas tous les citer, je nommerai toutefois M. BIBENT, responsable des chercheurs de l'IRETIIJ et M. SCHWARTZ , responsable de l'exploitation qui a toujours fait preuve de la plus grande compréhension en face de nos problèmes.

Madame BESSIERES et Monsieur BONNAUD ont grandement contribué à la réalisation matérielle de ce mémoire , c'est de tout coeur que je les remercie pour la rapidité et la qualité de leur travail.

- 0 - 0 - 0 - 0 - 0 - 0 -

I N T R O D U C T I O N

Les travaux qui font l'objet de cet exposé ont été entrepris en 1965. Le Professeur Pierre CATALA, de la Faculté de Droit de Montpellier avait dès cette date pressenti l'importance que pouvait prendre l'informatique dans le domaine juridique. Il n'existait alors que fort peu d'expériences connues dans ce domaine. Le Professeur CATALA décida tout de même de tenter quelque chose avec l'aide de son ami le Professeur FALGUEIRETTES, de la Faculté des Sciences de Montpellier, Directeur du laboratoire de Calcul et Traitement de l'Information.

Un tel projet, même au stade des études préliminaires, nécessitait une étroite collaboration entre Juristes et Informaticiens. Aussi une équipe de recherche fut constituée, formée par moitié de Juristes (le Pr. CATALA et M. BRUNET) et d'Informaticiens (le Pr. FALGUEIRETTES et moi-même). La parfaite entente et la plus efficace coopération qui régnèrent toujours à l'intérieur de cette équipe, même lorsque le nombre de membres augmenta, furent sans aucun doute un facteur décisif de notre réussite.

Les secteurs d'application de l'informatique au droit peuvent être divers. Mais il en est un qui semblait particulièrement intéressant, celui de la jurisprudence. Lorsqu'un juge-

ment est rendu, il l'est en fonction de la loi. Mais cette dernière ne peut pas tout prévoir et il faut tenir compte de toutes les circonstances particulières de l'affaire. Mais si certaines circonstances ne sont pas expressément prévues par la loi, il n'en demeure pas moins vrai qu'elles peuvent se retrouver dans plusieurs affaires, et il serait anormal que deux affaires mènent à des conclusions contradictoires. Aussi en plus de la loi, tout jugement tiendra compte des cas semblables déjà jugés. C'est la jurisprudence.

Si le principe en est simple, son application l'est moins. En effet, les jugements sont de plus en plus nombreux, leurs conclusions dispersées dans divers bulletins ou publications, quand ils sont au moins publiés... Il est donc de plus en plus difficile, long et aléatoire pour les magistrats et les praticiens saisis d'une affaire, de retrouver les précédentes décisions rendues dans des cas plus ou moins similaires. S'ils ne peuvent recourir à des auxiliaires spécialisés, ils doivent procéder eux-mêmes à une recherche manuelle à l'aide de tables analytiques. Mais le résultat dépend essentiellement du savoir-faire du chercheur, du temps disponible, de la qualité des tables et de leur contenu. Or, il est bien évident que plus le volume à traiter augmente, plus ces conditions tendent à se détériorer...

Les ordinateurs actuels offrent une grande capacité de stockage, une grande vitesse de traitement et une grande fiabilité. Il semble donc qu'ils soient parfaitement adaptés à

ce genre de travail, l'idée de départ étant qu'un ordinateur sera toujours capable de restituer ce qu'on lui aura fourni et qu'il est d'autre part capable de tout absorber... . Sur un plan plus technique, on pouvait penser que le problème ne consistait qu'en une application à ce nouveau domaine des procédés de recherche documentaire qui commençaient à être exploités, en particulier dans les secteurs scientifiques et techniques.

C'est cet aspect des choses qui nous fit contacter M. GUEDENEY et M. de DAIN de la Compagnie IBM FRANCE, spécialistes dans ce domaine. Eux-mêmes, à cette époque, travaillaient à l'élaboration du système documentaire SAGESSE qui donna naissance par la suite au système IRMS. Notre collaboration dura environ un an, ce qui explique certaines caractéristiques communes à leur système et au notre. Mais il nous est très vite apparu qu'il y avait une différence fondamentale entre les deux secteurs. En effet, un ouvrage ou un article scientifique ou technique, porte sur un sujet bien précis qui peut être parfaitement décrit à l'aide de quelques mots. Ces termes caractéristiques, le plus souvent appelés mots-clefs, sont en nombre assez restreints pour un secteur considéré. De plus, chacun d'entre eux a une signification parfaitement précise. Par contre en jurisprudence, il est très difficile d'extraire d'un jugement les mots-clefs caractéristiques.

Ce n'est en effet pas la matière qui importe le plus. Supposons par exemple, une affaire d'accident de la circulation ayant occasionné une blessure grave à un piéton ; l'accident

s'est produit en ville, le piéton traversait hors d'un passage clouté, mais le conducteur du véhicule était en état d'ébriété... Ce qui est intéressant, c'est de pouvoir déterminer le partage des responsabilités dans un accident de la circulation. Mais les conclusions dépendront étroitement des circonstances, ce qui fait que du point de vue juridique, seuls les jugements rendus dans des circonstances analogues auront valeur de précédent.

Ceci montre bien que les faits ont autant d'importance que la règle, que même des détails peuvent et doivent être conservés et surtout qu'il doit être possible de tout utiliser pour interroger les fichiers. L'exploitation doit donc pouvoir être faite avec une très grande souplesse et un maximum de précision. Ceci étant entrevu, il ne nous restait plus qu'à mettre sur pied un système sur mesure, répondant exactement à nos besoins. C'est ce que nous avons essayé de faire et ce à quoi nous pensons avoir abouti.

Chapitre I

EXPOSE DU PROBLEME

Pour pouvoir tirer des conclusions valables d'une telle expérience, il fallait avoir une source de documents suffisamment importante. Mais à ce premier stade il était hors de question de tout prendre. Notre choix s'est porté sur la Cour de Cassation et plus précisément sur les Chambres Civiles. Quelles furent les motivations de ce choix ? Elles furent doubles : (voir note p. 10)

1) - La Cour de Cassation est la plus importante des juridictions. De ce fait, les arrêts qu'elle rend sont de la plus haute importance du point de vue jurisprudentiel.

2) - Une grande partie de ces arrêts (environ les 2/3) sont publiés dans le Bulletin des Chambres Civiles de la Cour de Cassation. Il était donc facile de se procurer la documentation.

Une fois ce choix effectué, il était nécessaire de fixer le volume. Pour tester le fonctionnement élémentaire des programmes, sur un plan purement technique, un corpus très réduit peut à la rigueur suffire. Par contre, si on désire tester une méthode, il faut un ensemble suffisant. Ce minimum fut estimé à une année d'arrêts (environ 4.000 arrêts) pour les tous premiers tests, quatre années pour tester le système et s'entraîner à son utilisation et enfin huit à dix années pour pouvoir tirer des

conclusions valables. C'est donc en fin de compte sur un corpus de 35.000 à 40.000 arrêts qu'il fallait tabler. D'autre part, il nous semble que sur une juridiction, il est inutile de couvrir plus de 15 années et qu'il n'est pas forcément souhaitable de mélanger plusieurs juridictions dans le même fichier. Cela revient à dire que le maximum d'arrêts à traiter simultanément doit pouvoir être évalué entre 60.000 et 80.000. Malgré tout, il ne faut pas que ce maximum prévisible soit une limite absolue pour le système. C'est ainsi que nous verrons dans la suite qu'il existe dans notre système actuel deux limites indépendantes, une à 65.535 et l'autre à 80.000, mais l'une comme l'autre pourrait facilement être augmenté si le besoin s'en faisait sentir.

Le nombre global d'arrêts étant ainsi fixé, il y a lieu de déterminer le volume de chaque arrêt. Nous avons déjà dit dans l'introduction, qu'il était pratiquement impossible de réduire un arrêt à quelques mots-clefs. Malheureusement, il est tout aussi impensable d'utiliser le texte complet. Celui-ci est très nettement trop long, pas assez dense (les mots et même les phrases inutiles y foisonnent) et par suite pas assez précis. L'idéal serait d'en refaire la rédaction spécialement pour cette application. Ceci était irréalisable pour deux raisons :

1) - Le manque de temps. Avec au départ un seul juriste à plein temps, il aurait fallu plusieurs années avant de pouvoir commencer à tester.

PAR CES MOTIFS, et sans qu'il soit besoin de statuer sur le second moyen;

CASSE ET ANNULE le jugement rendu le 23 janvier 1962 par le Tribunal d'instance de Paris (17^e arrondissement); remet, en conséquence, la cause et les parties au même et semblable état où elles étaient avant ledit jugement, et, pour être fait droit, les renvoie devant le Tribunal d'instance de Paris (1^{er} arrondissement).

N° 62-11.884. *Pillas c/ dame Delagrée et autre.*

Président : M. Blin. — Rapporteur : M. Voulet. — Avocat général : M. Lindon. — Avocats : MM. Talamon et Lemaissier.

DANS LE MÊME SENS :

24 janvier 1963, *Bull.* 1963, II, n° 84 (2°), p. 62;

4 juillet 1963, *Bull.* 1963, II, n° 496, p. 372 et les arrêts cités.

N° 19

1° JUGEMENTS ET ARRÊTS. — Mentions. —
 Désignation de la juridiction ayant statué.
 — Indication suppléée par les autres mentions de la décision.

2° TESTAMENT. — Legs. — Légataire. — Désignation. — Œuvre charitable. — Œuvre non déterminée. — Recherche de l'intention du testateur.

3° APPEL. — Infirmité. — Motifs. — Réfutation des motifs du jugement entrepris. — Nécessité (non).

1° *Il ne saurait être fait grief à un arrêt de ne pas mentionner la Cour d'appel qui l'a rendu, dès lors que cette indication peut être aisément suppléée par les autres mentions de l'arrêt, qui précise qu'il a été rendu sur appel d'un jugement du Tribunal de la Seine, et qui comporte en marge la mention d'enregistrement à Paris.*

2° *C'est par une interprétation souveraine de la volonté d'un testateur qu'une Cour d'appel décide que l'administration de l'Assistance publique de Paris était la bénéficiaire d'un legs d'argent liquide destiné à « fonder un ou des lits pour tuberculeux, un ou des lits pour vieillards », en relevant que la succession avait été ouverte à Paris et le testament déposé chez un notaire de cette ville, et que, « dans le départe-*

ment de la Seine, les services d'assistance sont centralisés par l'Assistance publique ».

Et en statuant ainsi, les juges d'appel écartent nécessairement les moyens tirés de l'indétermination de la personne gratifiée et de l'interdiction des legs avec faculté d'élire.

3° *Dès lors qu'une Cour d'appel, se livrant à une interprétation souveraine de la volonté des parties, infirme le jugement déféré, elle n'a pas à s'expliquer sur les motifs sur lesquels le Tribunal s'était fondé pour admettre l'interprétation contraire.*

6 janvier 1965.

Rejet.

Sur le premier moyen :

Attendu qu'il fait grief à l'arrêt confirmatif attaqué de n'indiquer en aucune de ses parties, le nom de la Cour d'appel qui l'a rendu;

Mais attendu que l'arrêt précise qu'il intervient sur appel d'un jugement du Tribunal de la Seine et qu'en marge est inscrite la mention d'enregistrement à Paris, au bureau de la Cour d'appel, qu'ainsi l'indication omise peut être aisément suppléée par les autres mentions de l'arrêt; que le moyen n'est pas fondé;

Sur le second moyen, pris en ses diverses branches :

Attendu qu'il résulte des énonciations de l'arrêt infirmatif attaqué que la dame Ozonne, fille adoptive du sieur Meurice, décédée en 1939 sans postérité, a laissé des dispositions testamentaires où elle exprime notamment sa volonté que les legs particuliers qu'elles contiennent soient « exempts de tous droits à l'acquittement desquels serviront mes immeubles, 24, rue Fortuny, à Veules-les-Roses », et ajoute : « Si après tous ces dons il reste de l'argent liquide, fonder un ou des lits pour tuberculeux, un ou des lits pour vieillards, au nom de Paul Meurice »; que, par jugement du 1^{er} mars 1936, il a été fait droit à la demande des héritiers de deux autres filles adoptives, elles-mêmes héritières dudit sieur Meurice, tendant à l'exercice du retour légal de l'adoptant sur les deux immeubles;

Attendu que, sur la tierce opposition de l'Assistance publique à Paris, et par l'arrêt attaqué, la Cour d'appel a décidé que la dame Ozonne avait disposé de ses immeubles pour le paiement des frais afférents à chaque legs, et dit que l'Assistance publique était légataire du restant de l'argent liquide;

Attendu qu'il est fait grief à la Cour d'appel d'avoir procédé par simple affirmation en déclarant que « l'Assistance publique de Paris était la bénéficiaire éventuelle de ce legs particulier » de ne pas avoir réfuté en infirmant, les motifs du jugement que les consorts Meurice avaient faits leurs en concluant à la confirmation de ce jugement, notamment en ce qu'ils constataient que l'examen du testament ne permettait pas de décider que la *de cujus* avait voulu gratifier les pauvres de Paris plutôt que ceux de Chambéry, ville où elle avait des attaches, d'avoir omis de s'expliquer sur les griefs d'indétermination de la personne gratifiée et d'interdiction du legs avec faculté d'élire, alors enfin, qu'en définitive la seule question était de savoir si la testatrice avait disposé des immeubles

2) - L'ignorance de la façon de procéder. Nous ne savions pas, en effet, quel était le meilleur contenu et la meilleure structure pour un tel texte. C'est peu à peu, grâce à l'utilisation de ce système, que nous avons pu dégager un certain nombre de règles régissant la rédaction de ce qu'il est maintenant convenu d'appeler des abstracts.

Il était donc nécessaire d'utiliser ce que nous trouvions. Or dans le bulletin, chaque arrêt est accompagné de deux éléments (voir figure 1) :

1) - En gros caractères gras, un résumé très succinct, dit "titre principal". Ecrit en style télégraphique, il a une très haute densité d'information. Sa structure, en paragraphes et phrases, ces dernières souvent très courtes et même réduites à un seul mot, semble très propice au genre de traitement que nous envisagions.

2) - En caractères fins, un résumé beaucoup plus long. Ce dernier est caractérisé par l'absence de structure, étant parfois composé d'une seule phrase d'une demi-page ou plus !

Sa densité d'information et sa structure nous firent choisir le titre principal comme source de notre information. Nous verrons dans le chapitre sur les résultats obtenus, ce qu'il y a lieu d'en penser ; mais nous pouvons préciser de suite que si cette source ne saurait absolument pas suffire pour une exploitation réelle du système, elle nous a toutefois permis de tester parfaitement notre méthode.

Tout système informatique est partiellement indépendant du matériel utilisé. Néanmoins, sa réalisation pratique dépend du type d'ordinateur et de ses caractéristiques. Nous avons voulu faire ce système pour une installation moyenne en nous basant sur l'équipement de la Faculté des Sciences. Ce matériel était du reste seulement commandé lors du démarrage de nos recherches et ne fut livré qu'en octobre 1967. Il est composé :

1) - D'une unité centrale IBM 360/40 d'une capacité de mémoire centrale de 64 K.

2) - De trois unités de disques magnétiques 23II (capacité maximum de 7,5 millions de caractères par lecteur). Sur ces trois unités, une est utilisée par le système d'exploitation et seuls deux lecteurs sont utilisés par les fichiers "jurisprudence". En fait, on peut même utiliser un seul lecteur mais en réduisant un peu les possibilités.

3) - D'un lecteur-perforateur de cartes 2540.

4) - D'une imprimante I403/N1 (1100 lignes à la minute sur 132 caractères).

5) - D'une machine à écrire 1052 au pupitre.

La configuration assez réduite n'autorise qu'un seul système d'exploitation, de Disk Operating System (D.O.S.). Tous les programmes constituant notre ensemble de traitement tournent sous contrôle du DOS.

Le problème peut donc être à présent bien défini. Etant donné cette configuration machine et étant donné cette source d'information, il faut :

1) - Constituer sur disques magnétiques des fichiers tels qu'on puisse y retrouver toute l'information contenue dans les documents de départ, mais sous une forme permettant une consultation rapide et mettre au point un ensemble de programmes permettant la maintenance de ces fichiers, c'est-à-dire leur création, leur mise à jour et leur édition.

2) - Posséder un programme d'interrogation de ces fichiers. Ce dernier devra permettre de poser les questions soit par cartes, soit par le clavier de la machine à écrire. Ces questions seront posées à l'aide de termes et d'opérateurs logiques. C'est ainsi que si nous reprenons l'exemple de l'accident de la circulation, il faudra pouvoir demander quels sont tous les arrêts rendus dans lesquels on trouve dans une même phrase ACCIDENT ET CIRCULATION, dans une même phrase PARTAGE ET RESPONSABILITE, dans une même phrase HORS ET PASSAGE ET CLOUTE, dans un même paragraphe PIETON ET VICTIME, dans une même phrase CONDUCTEUR ET (IVRE OU EBRIETE).

La question étant ainsi posée, l'ordinateur pourra éventuellement "dialoguer" avec l'utilisateur en lui donnant des renseignements sur le résultat de sa recherche, et il devra de toute façon lui donner la ou les solutions qu'il aura trouvées. Ces résultats pourront consister soit en les références des arrêts, soit en les titres principaux de ces arrêts, cette sortie étant effec-

tuée sur l'imprimante. Un problème annexe, mais très important, est que cette solution puisse être obtenue dans le minimum de temps.

Note - Pour un exposé plus détaillé des choix effectués au départ, on consultera avec profit l'article du Pr CATALA "l'Informatique Appliquée à la Jurisprudence Civile" paru aux cahiers de l'Aupelf - juillet 1969

-o-o-o-o-

Chapitre II

COLLECTE DE L'INFORMATION DE BASE

FICHER CARTES DES ARRETS

Dans le chapitre précédent, nous avons vu que l'information de base était constituée par les "titres principaux" des arrêts publiés au Bulletin.

La Cour de Cassation est divisée en plusieurs chambres, de nombre variable suivant les années (en particulier, il a existé pendant un certain temps, une chambre temporaire des expropriations). Chacune de ces chambres est plus ou moins spécialisée. Certains jugements très importants sont même rendus par toutes les chambres réunies. Afin de systématiser la codification, nous avons considéré la réunion de toutes les chambres comme formant une chambre distincte. Ceci étant, pour chacune des années, nous avons numéroté les chambres de 1 à n, n étant variable suivant l'année, généralement 6 ou 7, dans l'ordre où elles apparaissent sur le Bulletin. Ceci est une convention presque à usage interne...

Sur le Bulletin, à l'intérieur de chaque chambre, les arrêts sont numérotés en séquence dans l'ordre chronologique. La référence d'un arrêt est donc constituée par l'année, la chambre et le numéro de séquence à l'intérieur de la chambre. Ce dernier

6510016320 DE FAUTES RETENUES EN PREMIERE INSTANCE/ARRET ECARTANT CERTAINES D EN
 6510016330TRE ELLES/MAINTIEN DE LA CONDAMNATION AU MEME MONTANT *
 6510017110060165*FONDS DE GARANTIE AUTOMOBILE/VVE MONTISCI*FONDS DE GARANTIE AUT
 65100171200MOBILE/CONDAMNATION DU FONDS AU PAYEMENT DE L INDEMNITE ALLOUEE A LA
 6510017130VICTIME OU A SES AYANTS-DROIT/ACCIDENT CAUSE PAR DES MEMBRES DES FORCE
 6510017140S ALLIEES EN FRANCE/RECOURS GRACIEUX PREALABLE AUPRES DE L ETAT D ORIG
 6510017150INE/NECESSITE NON *
 6510018110060165*PILLAS/DAME DELAGREE*GARANTIE/APPEL EN GARANTIE/EFFET/CONDAMNAT
 6510018120ION DE L APPELE EN GARANTIE AU PROFIT DU DEMANDEUR PRINCIPAL NON *
 6510019110060165*LANGLOIS BERTHELOT/L ASSISTANCE PUBLIQUE*JUGEMENTS ET ARRETS/RE
 6510019120NTIONS/DESIGNATION DE LA JURIDICTION AYANT STATUE/INDICATION SUPPLEEE
 6510019130PAR LES AUTRES MENTIONS DE LA DECISION*
 6510019210TESTAMENT/LEGS/LEGATAIRE/DESIGNATION/CEUVRE CHARITABLE/CEUVRE NON DETE
 6510019220RMINEE/RECHERCHE DE L INTENTION DU TESTATEUR*
 6510019310APPEL /INFORMATION/MOTIFS/REFUTATION DES MOTIFS DU JUGEMENT ENTREPRIS/N
 6510019320ECESSITE NON *
 6510020110060165*HACQUARD/CLEMENT*PROPRIETE/PREUVE/POSSESSION/CONFLIT DE POSSESS
 6510020120IONS/POUVOIR D APPRECIATION DES JUGES DU FOND*
 6510021110060165*DUNOYER DE SEGONZAC/ETAT FRANCAIS*SERVITUDES/SERVITUDE LEGALE/S
 6510021120ERVITUDE D UTILITE PUBLIQUE/SERVITUDE MILITAIRE/PLACE DE GUERRE/DECLAS
 6510021130SEMENT/MAINTIEN DE LA SERVITUDE NON-AEDIFICANDI/LOI DU 19 OCTOBRE 1919
 6510021140 PORTANT DECLASSEMENT DE L ENCEINTE DE LA PLACE DE LILLE*
 6510022110110165*FRANCOIS/VVE BLOUIN*ACTIONS POSSESSOIRES/CONCITIONS/POSSESSION/
 6510022120CARACTERES/ANNALITE/CONSTATATIONS SUFFISANTES*
 6510022210ACTIONS POSSESSOIRES/CONDITIONS/POSSESSION/CARACTERES/CARACTERES NON C
 6510022220ONTESTES/CONTESTATIONS SPECIALES/NECESSITE NON *
 6510023110110165*DAME MOHAMED BEN HAMOU BEN EL ADEL/MOHAMED BEN HADJ AISSA DIT K
 6510023120HALIFA*MAROC/IMMEUBLE/IMMEUBLE IMMATRICULE/DROIT DE PREEMPTION CHEFAA
 6510023130 /EXERCICE/PARTAGE ANTERIEUR/EFFET DECLARATIF/PUBLICATION AUX LIVRES F

Figure 2. LISTING DES DOCUMENTS DE BASE.

reprend à 1 au début de chaque année. Sa valeur maximum est voisine de 1300, ce qui oblige à prévoir 4 chiffres pour ce numéro.

En plus de cette référence, deux autres renseignements peuvent servir à caractériser un arrêt ; ce sont la date du jugement et le nom des parties. Nous serons obligés de garder ces deux éléments si nous désirons pouvoir retrouver un arrêt dans une publication autre que le Bulletin (Dalloz, Gazette du Palais, J.C.P. , etc...). Ces publications, en effet, ne font pas mention du numéro de l'édition du Bulletin.

Le fichier de départ est perforé sur cartes, directement à partir du Bulletin, sans aucune préparation. Le dessin des cartes est le suivant (voir figure 2, un "listing" de ce fichier) :

Colonnes 1 et 2	: l'année (les deux derniers chiffres)
Colonne 3	: le numéro de la chambre
Colonnes 4 à 7	: le numéro de séquence de l'arrêt
Colonne 8	: le numéro de paragraphe. Certains arrêts présentent plusieurs aspects, chacun d'eux faisant l'objet d'un paragraphe distinct.

Pour l'instant, sur dix ans, nous n'avons pas trouvé un seul arrêt ayant plus de 9 paragraphes. Il est bien évident que si cela se présentait un jour, il suffirait d'attribuer des lettres comme numéro de paragraphe.

Colonne 9 et 10	: le numéro de la carte à l'intérieur du paragraphe.
-----------------	--

Ce numéro est un simple numéro de séquence attribué par la perforatrice. La aussi nous aurions sans doute pu nous contenter d'un seul chiffre, un paragraphe ne dépassant pratiquement jamais 9 cartes. Malgré tout, nous avons préféré en garder deux pour deux raisons :

1) - Si nous voulons appliquer notre système à des textes autres que les titres principaux, nous ne sommes pas certains de pouvoir faire tenir un paragraphe sur 9 cartes au plus.

2) - 9 cartes nous suffisant pour le moment, nous en profitons pour faire faire la numérotation de 10 en 10 ce qui permet d'effectuer facilement des corrections par insertion de cartes supplémentaires.

Ces 10 premières colonnes ont la même structure pour toutes les cartes d'un arrêt.

Dans les colonnes 11 à 16 de la première carte de chaque arrêt, on trouve la date du jugement (jour, mois, an) qui est suivie en colonne 17 par un astérisque. Ce dernier sert uniquement à des vérifications de cadrage. S'il ne se trouve pas dans cette colonne, c'est que la référence a été allongée ou raccourcie. Comme cette référence est reproduite automatiquement dans les cartes qui suivent, il y a lieu de s'assurer de son exactitude.

Toujours dans cette première carte, à partir de la colonne 18, on trouve les noms des parties, séparées par une barre

de fraction. Cette zone peut déborder de la première carte et continuer sur la seconde, voire même les suivantes. Sa limite est indiquée par un astérisque. On doit noter de suite que ces deux renseignements (date et noms des parties) ne se trouvent qu'en tête du premier paragraphe et ne sont pas répétés en tête des suivants.

Derrière l'astérisque limitant les parties commence le texte proprement dit avec les conventions suivantes :

1) - Un mot peut être coupé en deux pourvu que la dernière lettre de sa première moitié soit en colonne 80 d'une carte et la première lettre de sa seconde moitié en colonne 11 de la carte suivante.

2) - A l'intérieur d'une même phrase, les mots doivent être séparés par au moins un espace. Si on le désire, on peut en mettre plusieurs sans inconvénient (en particulier pour éviter de faire commencer un mot en colonne 80 d'une carte ou lors de corrections pour le remplacement d'un mot par un plus court).

3) - Les phrases sont séparées par des barres de fraction.

4) - La fin d'un paragraphe est signalée par un astérisque. Les colonnes situées derrière cet astérisque ne doivent pas être utilisées et le paragraphe suivant doit commencer sur la carte suivante.

Cette structure et ces conventions relativement simples ont permis de faire effectuer la perforation directement à par-

tirtir du Bulletin sans aucune préparation par des perforatrices "standard". Nous devons à la vérité de dire qu'il leur faut tout de même un certain temps d'adaptation au langage juridique et que les premiers essais ont révélé un nombre d'erreurs non négligeable.

Un problème particulier qui dut être résolu très vite est celui des abréviations. Il n'en existe pratiquement aucune dans ces textes et c'est ainsi qu'on trouvera toujours en toutes lettres PRESIDENT DIRECTEUR GENERAL et non pas PDG , ou SOCIETE NATIONALE DES CHEMINS DE FER FRANCAIS et non pas SNCF. Or, il est tout de même important de pouvoir les condenser. Cette opération a été laissée à la discrétion des perforatrices qui s'en tirent fort bien. De même, il leur a été demandé de relier par un trait d'union les mots de certaines expressions indissociables, en particulier les expressions latines telles que IN-LIMINE-LITIS ou PRETIUM-DOLORIS.

Ce sont, en particulier, ces libertés et initiatives laissées aux perforatrices qui nous ont poussés à ne pas faire vérifier ces cartes (nous entendons bien vérifier à l'aide de vérificatrices). Or, il est inévitable qu'il y ait des erreurs. Pour les détecter, nous possédons deux moyens :

1) - Lorsqu'une certaine quantité de cartes a été perforée, nous en effectuons un listing. La simple lecture de ce listing permet d'éliminer les erreurs les plus grossières. Mais l'expérience nous a montré que certaines erreurs, telles que l'inversion de deux lettres dans un mot sont très difficiles à détecter

ainsi.

2) - Lors de la phase d'analyse de ces textes, l'ordinateur éditera les anomalies, qu'elles soient de structure ou de vocabulaire. Ceci permet alors de remettre parfaitement en ordre ce fichier.

La seule erreur qui soit pratiquement indétectable est l'omission d'un paragraphe ou l'attribution d'un paragraphe d'un arrêt a un autre... .

Sur le plan volume, avec les seuls titres principaux comme source de l'information, on a en moyenne 4 cartes par arrêt. Compte tenu des lacunes présentées par ces textes, il faudrait normalement tabler sur 6 à 7 cartes pour un fichier réellement opérationnel.

-O-O-O-O-O-

Chapitre III

ANALYSE DES DOCUMENTS DE BASE

LES FAMILLES DE PROGRAMMES D1, D2 ET D3

Lorsqu'on envisage d'effectuer de la recherche documentaire, on peut utiliser deux techniques. La première consiste à garder les documents sous la forme d'origine et à "balayer" l'ensemble de tous ces documents à chaque interrogation. Si on désire une comparaison, c'est le cas de quelqu'un qui cherche un article dans un catalogue qui n'est pas muni d'un index ; la seule solution consiste alors à parcourir tout le catalogue... .

Ce procédé va bien entendu être très long de l'interrogation. Il ne faut pas penser pour autant qu'il est inapplicable. Dans la mesure où on possède une très grosse mémoire centrale permettant de traiter un grand nombre de questions simultanément, dans la mesure également où on a effectivement un grand nombre de questions à traiter à la fois, chacune de ces questions ayant un grand nombre de solutions, on peut avoir intérêt à utiliser une telle technique. De plus, un tel procédé présente un avantage non négligeable pour la mise à jour des fichiers. En effet, l'introduction de nouveaux textes se bornera à les mettre à la suite de ce qui existe déjà, la suppression, le remplacement ou la modification d'un texte déjà existant ne po-

sant guère plus de problème. De tels fichiers, enfin, ayant un accès purement séquentiel, peuvent être placés sur bandes magnétiques, ce qui résoud le problème du volume total... .

A l'opposé, la deuxième technique possible consiste à utiliser un "fichier inversé". Pour reprendre la comparaison de ci-dessus, un tel fichier serait l'index du catalogue. On y trouve répertorié dans un certain ordre tous les articles contenus dans le catalogue, chacun d'entre eux étant accompagné seulement du numéro des pages où il apparaîtrait.

Il est évident que dans le cas de la simple consultation d'un catalogue, cette dernière méthode est de beaucoup la plus rapide. Dans le cas d'une recherche documentaire complexe, le nombre d'opérations à effectuer va être assez important. Il va falloir en effet, rechercher les index de tous les termes utilisés dans le corps de chaque question et combiner entre eux leur contenus pour obtenir finalement les seules références des solutions. Pour peu que le nombre de références associées à chaque terme soit très élevé, le temps de recherche peut se trouver fortement augmenté... .

Toutefois, dans le cas qui nous intéresse, c'est-à-dire, une recherche portant sur relativement peu de questions à la fois, mais sur un gros corpus, chaque question ne devant avoir probablement que peu de solutions, la solution du fichier inversé semble être de loin la meilleure. Et nous pensons qu'elle le reste en dépit des difficultés qu'elle présente pour sa mise en oeuvre. C'est la solution que nous avons utilisée.

6510006110040165*TREGUAI/DEMOISELLE SEVESTRE*MANDAT/ETENDUE/DEPASSEMENT/MANDAT A
6510006120PPARENT*

6510232110300365*LELUNG/CIE D ASSURANCES LABEILLE*MANDAT/ETENDUE/DEPASSEMENT/MA
6510232120NDAT APPARENT*

6610211110280366*VEUVE D ESTIENNE D ORVES/VILLE D AIX EN PROVENCE*CASSATION/MOYE
6610211120N NUVEAU/MANDAT APPARENT/CROYANCE LEGITIME DU TIERS*
6610211210 ACTION EN JUSTICE/DEFENSE/RESISTANCE INJUSTIFIEE/FAUTE/CONSTATATIONS
6610211220SUFFISANTES*

6610488110021166*BARKER /YUNG*CASSATION/MOYEN NOUVEAU/MANDAT/MANDAT APPARENT/MAND
6610488120AT VERITABLE SEUL INVOQUE DEVANT LES JUGES DU FOND*

6610556110201266*SOCIETE NELLYS ET AUTRES/MEGRET*JUGEMENTS ET ARRETS/CONCLUSIONS
6610556120/OBLIGATION DE JUGER DANS LEURS LIMITES/MANDAT/MANDAT APPARENT/MANDAT
6610556130VERITABLE SEUL INVOQUE*

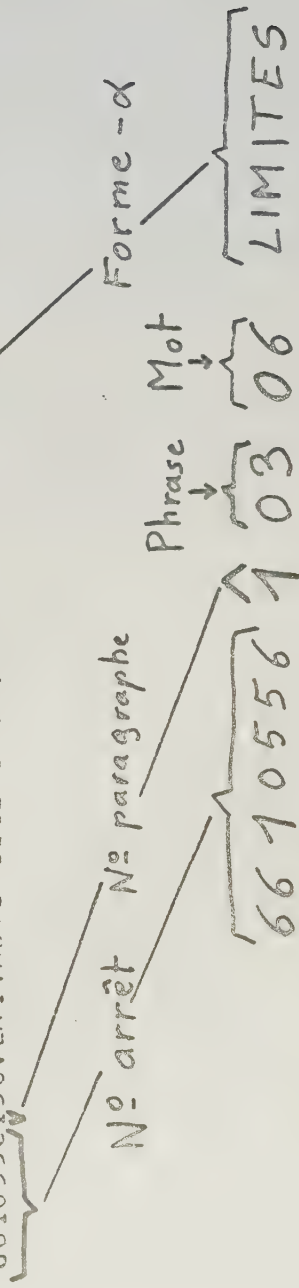


Figure 3. Extraction d'une forme-α et constitution de sa localisation.

La création du fichier inversé, pièce maitresse de la recherche, va donc nécessiter l'extraction de chaque terme contenu dans les textes d'origine, chaque terme extrait devant être accompagné de l'information suffisante pour retrouver d'où il provient. Cette information associée sera appelée dans ce qui suit, la "localisation".

Son contenu est le suivant : (voir figure 3)

- référence complète de l'arrêt : année, chambre et N°
- numéro de paragraphe
- numéro de la phrase à l'intérieur du paragraphe
- numéro de mot à l'intérieur de la phrase.

L'extraction des termes et la fabrication de leur localisation va être le travail principal de la famille de programmes D1 (fig. 5). Le premier de la famille, le plus simple, porte le numéro D1. Il admet en entrée un fichier sur cartes avec la structure décrite dans le chapitre précédent. Ce fichier est supposé classé, au moins à l'intérieur de chaque arrêt. Pour chaque arrêt, le programme D1 effectue à partir de la colonne 18 de la première carte une recherche du prochain astérisque qui indique la fin des parties. Au-delà, commence le texte qui va être "démonté". Les compteurs de phrase et de mot étant mis à 1, on va chercher où commence (il peut être précédé d'un nombre variable d'espaces) et où se termine le premier terme (à la rencontre d'un espace, d'une barre de fraction ou d'un astérisque). Les références de l'arrêt et du paragraphe sont alors extraites du début de la carte. Ce premier terme accompagné de sa localisation

est alors écrit sur disque dans un fichier séquentiel à enregistrements de longueur fixe et bloqués. A ce stade tout est conservé sous forme alphanumérique. Le processus continue alors avec le terme suivant après avoir augmenté de 1 le compteur de mots, sauf si le délimiteur était une barre de fraction (cest alors le compteur de phrase qui doit être augmenté de 1, celui de mots étant remis à 1) ou un astérisque (les deux compteurs doivent être remis à 1 puisqu'il s'agit d'un changement de paragraphe ou d'arrêt).

Nous venons d'indiquer un peu plus haut que les enregistrements du fichier de sortie sont de longueur fixe. Or, les termes qui sont extraits, formés d'une chaîne de caractères quelconques placés entre deux délimiteurs et que nous appellerons dans la suite forme alphanumérique (forme- α), sont de longueur variable. Les enregistrements de longueur variable étant acceptés sur disque, il aurait semblé préférable de les utiliser. Le seul ennui est que la prochaine étape du traitement va consister en un tri portant sur la forme- α elle-même et que le tri standard ne permet pas d'utiliser des arguments de tri de longueur variable. Nous avons donc arbitrairement fixé la longueur à 25 caractères, étant entendu que les formes- α sont cadrées à gauche et complétées à droite par des espaces ou éventuellement tronquées à droite. Même parmi les expressions indissociables obtenues en liant plusieurs formes- α par des traits d'union, il y en a fort peu qui sont ainsi tronquées.

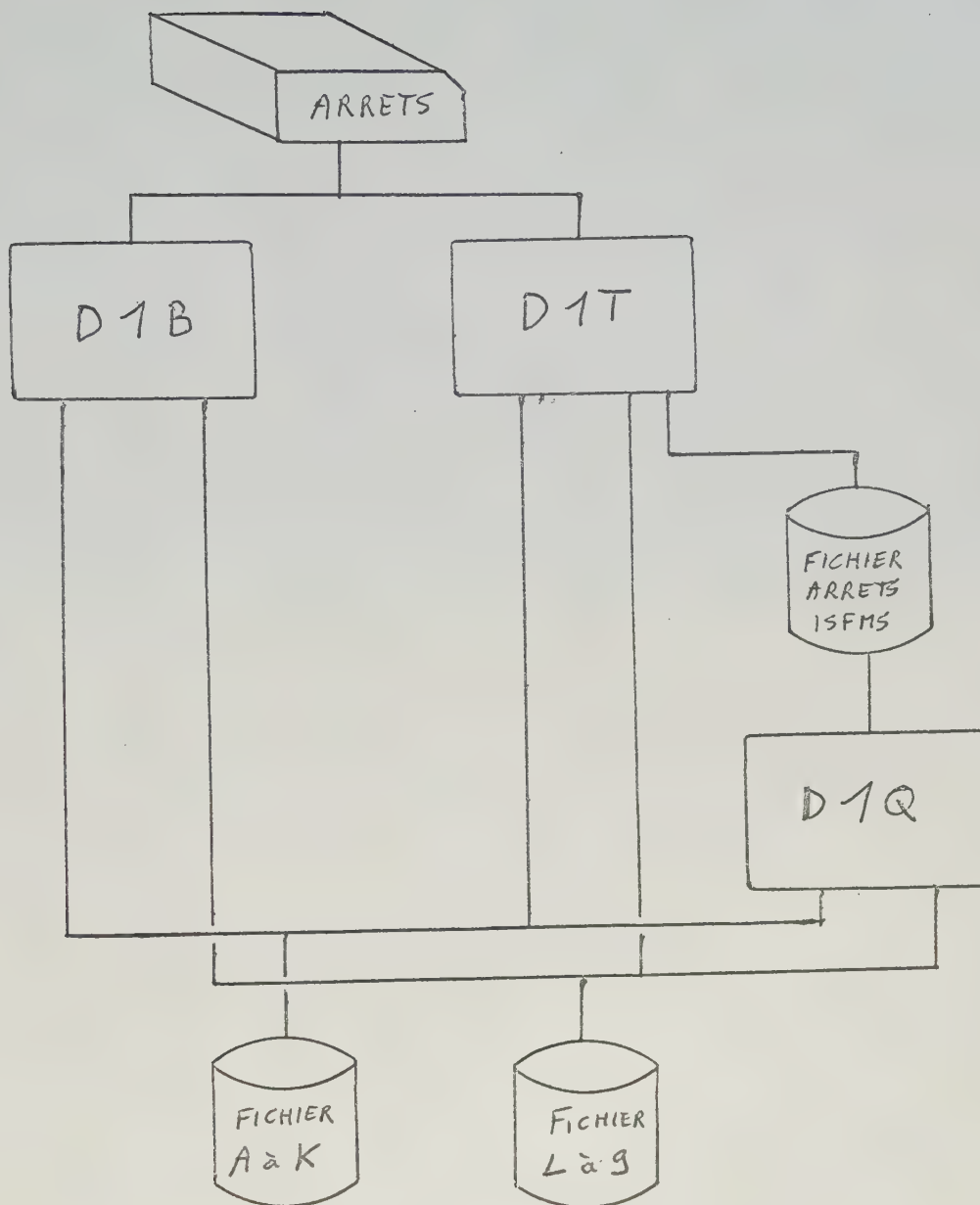


Fig - 4 -

Ce premier programme D1 lit donc un fichier sur cartes et fabrique un fichier sur disque. Mais très vite est apparue une première difficulté. Le programme de tri standard nécessite un fichier de travail sur disque d'un volume double de celui du fichier à trier. Or, il est intéressant de pouvoir effectuer le traitement par gros morceaux, pratiquement par année entière. Le fichier de sortie qui en résulte risque d'être trop important. Aussi il a été procédé à une première modification du programme qui a consisté à utiliser en sortie deux fichiers sur disque, un pour les formes- α commençant par une des lettres de A à K et l'autre pour les autres. Ce programme là reçut le nom de D1B.

Nous avons déjà dit au cours du chapitre 2, qu'un certain nombre d'erreurs présentes dans le fichier sur cartes n'étaient détectées que lors de la phase d'analyse. Ceci implique donc que l'incorporation d'une nouvelle année au corpus déjà existant ne pourra jamais se faire en un seul passage et qu'elle nécessitera pratiquement toujours une séance de corrections, voire deux ou trois. Tel qu'il est le programme D1B ne permet qu'une procédure : la correction du fichier cartes et la reprise intégrale de son traitement. L'inconvénient est d'être obligé de faire repasser toutes les cartes dans le lecteur. Aussi, nous a-t-il paru plus astucieux de mettre au point deux autres versions du programme D1. Ce sont les programme D1T et D1Q.

D1T accepte en entrée le même fichier cartes, fabrique en sortie les deux mêmes fichiers que D1B, mais fabrique en outre

ERREUR SEQUENCE - NON CHARGEE	6510013110050155*HERVET/QUILLIVIC*MEDECIN CHIRURGIEN/HONORAIRES/PAYEMENT/MEDECIN 651001432
ERREUR SEQUENCE - NON CHARGEE	6510019120NTIONS/DESIGNATION DE LA JURIDICTION AYANT STATUE/INDICATION SUPPLEEE 651002112
ERREUR SEQUENCE - NON CHARGEE	6510019130PAR LES AUTRES MENTIONS DE LA DECISION* 651002112
ERREUR SEQUENCE - NON CHARGEE	6510019210TESTAMENT/LEGS/LEGATAIRE/DESIGNATION/OEUVRE CHARITABLE/OEUVRE NON DETE 651002112
ERREUR SEQUENCE - NON CHARGEE	6510019220RMINEE/RECHERCHE DE L INTENTION DU TESTATEUR* 651002112
ERREUR SEQUENCE - NON CHARGEE	6510019310APPEL/INFORMATION/MOTIFS/REFUTATION DES MOTIFS DU JUGEMENT ENTREPREIS/N 651002112
ERREUR SEQUENCE - NON CHARGEE	6510019320NECESSITE NON * 651002112
ERREUR SEQUENCE - NON CHARGEE	6510020110060155*HACQUARD/ CLEMENT*PROPRIETE/PREUVE/POSSESSION/CONFLIT DE POSSESS 651002112
ERREUR SEQUENCE - NON CHARGEE	6510020120NTIONS/POUVOIR D APPRECIATION DES JUGES DU FOND* 651002112
ERREUR SEQUENCE - NON CHARGEE	65100211100260165*DUNCYER DE SECONZAC/ETAT FRANCAIS*SERVITUDES/SERVITUDE LEGALE/S 651002112
ERREUR SEQUENCE - NON CHARGEE	6510028120OEURS BLESSES PAR LA PROJECTION DU CORPS D UN AUTRE VOYAGEUR HEURTE PAR 651003312

Figure 6 - Sortie des erreurs lors du passage de DIT. La colonne de droite donne la référence de la dernière carte acceptée.

un troisième fichier sur disque. Ce dernier est tout simplement composé d'images des cartes organisées en séquentiel indexé avec blocage, les 10 premières colonnes de chaque carte servant de clef. Une zone d'overflow est réservée sur chaque cylindre plus une zone indépendante en fin de fichier. Si des cartes se présentent déclassées (elles doivent être introduites après avoir été triées sur les 10 premières colonnes) ou avec les mêmes 10 premières colonnes que la carte précédente, elles ne sont pas prises en compte et listées sur l'imprimante avec l'indication de la dernière carte chargée (voir figure 6). Ceci permet de détecter déjà un certain nombre d'erreurs.

Le dernier de cette famille, D1Q est semblable à D1B à ceci près qu'il n'utilise plus en entrée un fichier sur cartes, mais le fichier séquentiel indexé sur disque fabriqué par D1T, ce dernier fichier étant exploité en séquentiel (fig. 4).

Bien entendu, entre l'exploitation de D1T et celle de D1Q, il y aura eu un passage de corrections. Pour ce faire, nous disposons de deux programmes, EDITAR et MAJAR.

EDITAR est un simple programme d'édition qui, lisant en séquentiel le fichier séquentiel indexé, liste son contenu sur l'imprimante.

MAJAR permet d'effectuer des adjonctions ou des modifications à ce fichier. Il accepte en entrée un fichier sur cartes, cartes de dessin analogue à celui du fichier général des arrêts. Si la référence d'une carte (c'est-à-dire, le contenu de

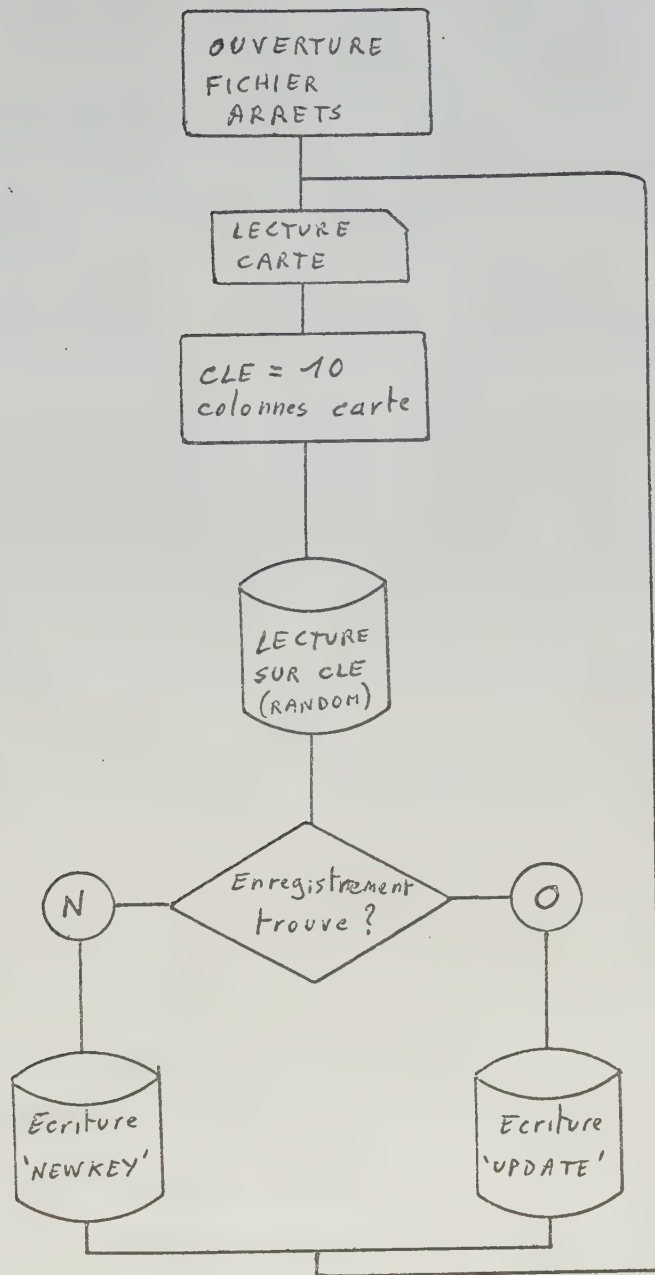


Fig - 7 - MAJAR - Principe général -

ses 10 premières colonnes) n'existe pas encore dans le fichier disque, elle y est ajoutée. Si par contre cette référence existe, le contenu de la carte est transféré sur le disque au même emplacement (fig. 7).

A la suite d'un des programmes de la famille D1, on utilise toujours un programme de tri standard. Ce dernier accepte en entrée un des fichiers sur disque formé par les formes- α et leurs localisations, ordonne cet ensemble dans l'ordre alphabétique croissant et le replace dans la même zone de disque, avec la même structure. Ce nouveau fichier, qui comprend toujours un enregistrement par localisation pour chacune des formes- α , les présente à présent regroupées les unes à la suite des autres.

Comme nous le verrons dans le chapitre suivant, notre système comporte un "lexique", fichier sur disque comportant un enregistrement pour chacune des formes- α qui ont été rencontrées depuis le début. La constitution de ce lexique nécessite la connaissance des formes- α présentes dans les textes. C'est le rôle du programme D2 de les faire connaître. Ce programme accepte en entrée les fichiers formes- α + localisations obtenus par un des programmes de la série D1, ces fichiers ayant entre temps été triés sur la forme- α . Un tel fichier présente donc à la suite toutes les localisations de chaque forme- α . Le programme D2 compte le nombre de localisations de chacune et édite la forme accompagnée du nombre de localisations. Ce dernier renseignement est très utile car il permet de se faire une idée de

1 EMANANT
 6 EN
 1 ERREUR
 3 ET
 2 ETAT
 2 ETRANGER
 2 ETRANGERE
 2 EVALUATION
 2 EXPEDITION
 1 EXPULSION
 3 FAUTE
 1 FILIATION
 1 FONDS
 2 GENERAL
 1 HERITIER
 1 IMMEUBLES
 1 IMMOBILIER
 1 INJUSTIFIEE
 1 INVOQUE
 1 JOUR
 1 JOURNAUX
 1 JUDICIAIRE
 1 JUGEMENTS
 2 JUSTICE
 6 L
 3 LA
 1 LE
 1 LIEU
 2 LIQUIDATION
 1 LOCATION
 3 LOI
 1 LOIS
 1 MARI
 1 MARIAGE
 1 MOYEN
 1 MOYENS
 1 NATURELLE
 1 NECESSAIRES
 1 NECESSITE
 2 NON
 1 NOTAIRE
 1 NOUVEAU
 2 OBJET
 1 OMISSION
 2 PAR
 1 PARTIES
 1 PATERNITE
 1 PETITA
 1 PIECES
 1 PIGNORATIF
 1 PLAN
 4 PORTEE
 1 POSSESSION
 1 POSSESOIRES
 2 PRET

Fig. 8. Exemple de sortie du programme D2
 (appliqué à un fichier d'essai très réduit).

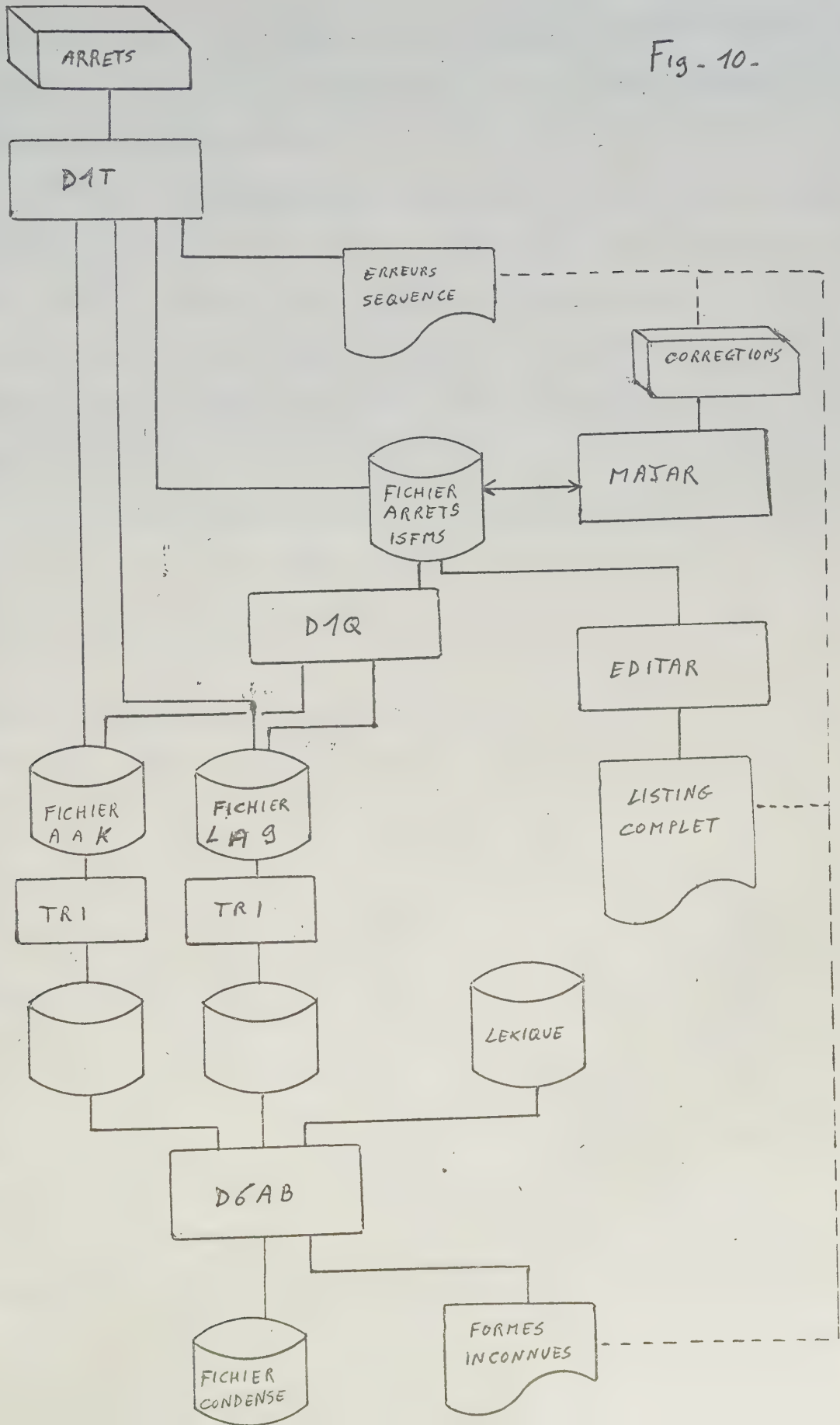
la valeur documentaire de telle ou telle forme qui a tendance à décroître si la forme est trop fréquente. En plus, à la fin de cette édition, on trouve le nombre total de formes- α rencontrées ainsi que le nombre de celles qui étaient distinctes (fig. 8).

Si ce programme permet de connaître le contenu des textes en vue de la constitution du lexique, il ne permet pas aisément de le compléter. Aussi il existe une autre version, portant le nom de D2B qui accepte aussi en entrée le lexique. Sa sortie n'est alors constituée que par les seules formes- α non encore incorporées au lexique, chacune d'elle étant accompagnée de son comptage. Les renseignements statistiques fournis à la fin portent sur l'ensemble des formes- α (présentes ou non au lexique) et sur celles qui ne s'y trouvent pas.

Enfin, cette première série de programmes est complétée par celui portant le nom de D3. Il accepte en entrée le même fichier que D2, mais il édite chacune des formes- α avec toutes ses localisations (fig. 9). Cette sortie très lourde n'est plus utilisée à présent et n'a servi qu'à effectuer manuellement une simulation de recherche.

Le programme D2B éditant les formes- α inconnues du lexique édite nécessairement les formes- α à l'orthographe erronée. De plus, une erreur de structure dans les cartes (absence d'un astérisque, perte d'une carte, etc...) se traduit presque inmanquablement par la sortie de formes- α inconnues (mots tronqués, dates, nom des parties, etc...). Le cycle de corrections

Fig - 10 -



correspond alors à la figure 10, le programme D2B étant en général remplacé par le D6AB qui donne en plus les localisations (ce programme sera vu plus loin).

Les programmes de la série D1, de même que ceux de la série D2 ainsi que D3 et D6AB ne réclament qu'une partition de 10 K. Le tri standard peut lui aussi théoriquement passer dans une petite partition, mais sa vitesse d'exécution est variable en fonction de la place disponible. Pour cette raison, il est préférable de lui laisser au moins 50 K.

Cette première phase du travail utilise un certain nombre de périphériques. Dans le tableau ci-dessous, on a placé XXX pour indiquer l'utilisation.

Programme	Lect.cartes	Imprim.	Disque 1	Disque 2
D1	XXX		XXX	
D1B	XXX		XXX	
D1T	XXX	XXX	XXX	XXX
D1Q			XXX	XXX
EDITAR		XXX		XXX
MAJAR	XXX			XXX
TRI			XXX	XXX
D2		XXX	XXX	
D2B		XXX	XXX	XXX
D3	XXX	XXX	XXX	
D6AB		XXX	XXX	XXX

Pour le traitement d'un fichier de 15.000 cartes environ, les deux fichiers de sortie de DIT comportant en moyenne 45.000 enregistrements chacun, les temps sont les suivants :

DIT 20 minutes, 10 minutes pour chaque tri, 5 minutes pour D6AB, ce qui donne 45 minutes ($20 + 2 \times 10 + 5$) pour la chaîne DIT, deux tris et D6AB. D1Q ne nécessite que 5 minutes, ce qui donne 30 minutes comme temps pour D1Q, deux tris et D6AB.

-o-o-o-o-

Chapitre IV

LE LEXIQUE

Sur un plan théorique, rien n'oblige à constituer un lexique qui regroupe toutes les formes- α rencontrées dans les textes utilisés. Il est, en effet, évident que le fichier inversé peut être parfaitement constitué sans qu'il y soit fait appel et l'interrogation elle-même peut parfaitement utiliser les formes- α dans le corps des questions.

Pourtant il nous a très vite paru utile de constituer un lexique. La principale raison fut d'avoir ainsi la possibilité de résoudre le cas des synonymes vrais. Une notion bien précise peut bien souvent être exprimée sous plusieurs formes différentes.

C'est ainsi qu'on peut utiliser indifféremment EPOUX ou MARI. Mais ce problème déjà grave empire du fait que pour nous (ou plutôt pour l'ordinateur) les différentes formes grammaticales du même mot sont considérées comme des formes- α distinctes.

La solution de ce dernier cas pouvait bien sûr être trouvée par d'autres procédés. Entre autres, nous pouvons citer :

1) - Troncation de toutes les formes- α à un nombre de lettres relativement faible . Les différentes formes grammaticales

les du même mot ne varient, en effet, généralement que sur les dernières lettres. Mais si le nombre de lettres conservées est fixe, ou il est élevé, et les mots courts ne bénéficient pas du procédé, ou il est faible, et il devient impossible de distinguer certains mots. Par exemple, si le nombre de lettres est inférieur à 7, il est impossible de séparer ACCORD de toutes les formes du verbe ACCORDER.

2) - Réduction automatique, lors de la phase d'analyse, de toutes les formes- α à une forme- α "racine" par application d'une sorte de programme d'analyse grammaticale. Ce serait peut-être la meilleure solution, si la langue française ne présentait pas autant de cas particuliers... . Le recensement de tous ces cas et leur traduction en algorithmes plus ou moins généraux ou en tables aurait été sans doute un travail tout aussi long que la constitution du lexique... .

3) - Ne rien faire lors de la phase d'analyse, mais reporter le problème lors de l'interrogation. On peut alors envisager d'interroger avec non pas des formes- α mais des racines en spécifiant par quoi elles peuvent être complétées. Ceci peut permettre de mettre en oeuvre un système plus rapidement, mais il demande plus de travail à son utilisateur futur. Ce dernier risque alors d'oublier une terminaison (ou un synonyme à la racine complètement distincte) et par suite d'avoir du "silence". Or, il est bien évident que ce dernier est beaucoup plus dangereux que le "bruit"... .

Si par contre, on répertorie toutes les formes- α existant dans les textes et qu'on associe à chacune d'elle une "notion", il ne restera plus qu'à utiliser ces notions pour l'interrogation et on sera certain de récupérer tous les synonymes vrais (dans la mesure où on leur aura associé la même notion), qu'ils soient de même racine ou non. Bien sûr, la constitution de ce lexique est une opération longue et laborieuse, mais nous avons estimé que les avantages retirés la justifient.

Nous aurions pu caractériser les notions par une des formes- α associées, par exemple EPOUX, MARI, etc... par MARI. Mais un tel procédé nous donnait des références de longueur variable, ce qui n'est guère pratique. Nous avons donc préféré utiliser des numéros. Là encore, nous avons le choix entre deux options : la distribution des numéros devait-elle matérialiser une structure logique à l'intérieur du lexique (par exemple une structure en arbre entre les notions, allant du général au particulier) ou au contraire ne rien représenter ? La première option aurait sans aucun doute offert des possibilités supplémentaires au moment de l'interrogation des fichiers. Mais en contrepartie elle nécessitait un travail préliminaire très ardu et une étude très fine du vocabulaire. Il n'est pas impossible que dans une étape ultérieure un tel travail soit entrepris, mais uniquement après qu'une assez longue expérience nous ait permis de bien connaître le vocabulaire utilisé.

La solution retenue fut donc la plus simple. Une première liste de formes- α ayant été obtenue en exploitant une seu-

le chambre sur une seule année, les numéros ont été attribués dans l'ordre croissant à ces termes eux-même classés par ordre alphabétique. Bien entendu, lorsqu'une des formes- α était synonyme d'une autre déjà traitée, le même numéro lui était attribué. De même, en trouvant dans la liste une des formes grammaticales d'un mot, nous avons introduit dans le lexique les autres formes.

De ce qui précède, on peut en déduire que nous avons banni du lexique toute idée de structure. Ce n'est pas exactement vrai. En effet, il existe des termes qui, sans être synonymes vrais, sont tout de même assez proches pour pouvoir, sinon recouvrir la même notion stricte, tout au moins représenter la même idée générale. C'est le cas par exemple des deux mots COMMERCE et COMMERCANT. Il peut être utile dans certains cas de pouvoir interroger les fichiers au niveau de cette notion plus générale. Pour ce faire, nous avons utilisé des numéros de notion à 5 chiffres, dans lesquels les 4 premiers sont caractéristiques de la notion générale, les notions particulières étant distinguées par le dernier chiffre. C'est ainsi que DISPONIBLE correspond à la notion 26250 et DISPONIBILITE à 26251. Le lexique est donc constitué, sinon d'un arbre unique, tout au moins d'une série de petits bouquets.

Ces règles étant posées, il reste à résoudre trois difficultés : le cas des termes sans signification, celui des formes numériques pures et enfin celui des homonymes. Voici ce qui

a été fait :

1) - Termes sans signification. Il s'agit soit des termes de liaison tels que les articles (LE, LA, etc ...), les prépositions (DU, DE, etc ...), d'autres termes analogues et même des termes trop fréquents ne signifiant plus rien sur le plan documentaire (toutes les formes du verbe ETRE, le mot ARTICLE, etc ...). La phase préliminaire d'analyse des textes ne les a pas écartés et en dépit du style des textes utilisés, on a une proportion de l'ordre de 40 % de ces mots. Tous ont reçus le même numero de notion "NUL". Ceci signifie que tous ces termes se trouvent dans le lexique mais accompagnés des 5 caractères "NUL" à la place des 5 chiffres du N° de notion. Ceci permet alors de les éliminer lors de la constitution du fichier inversé.

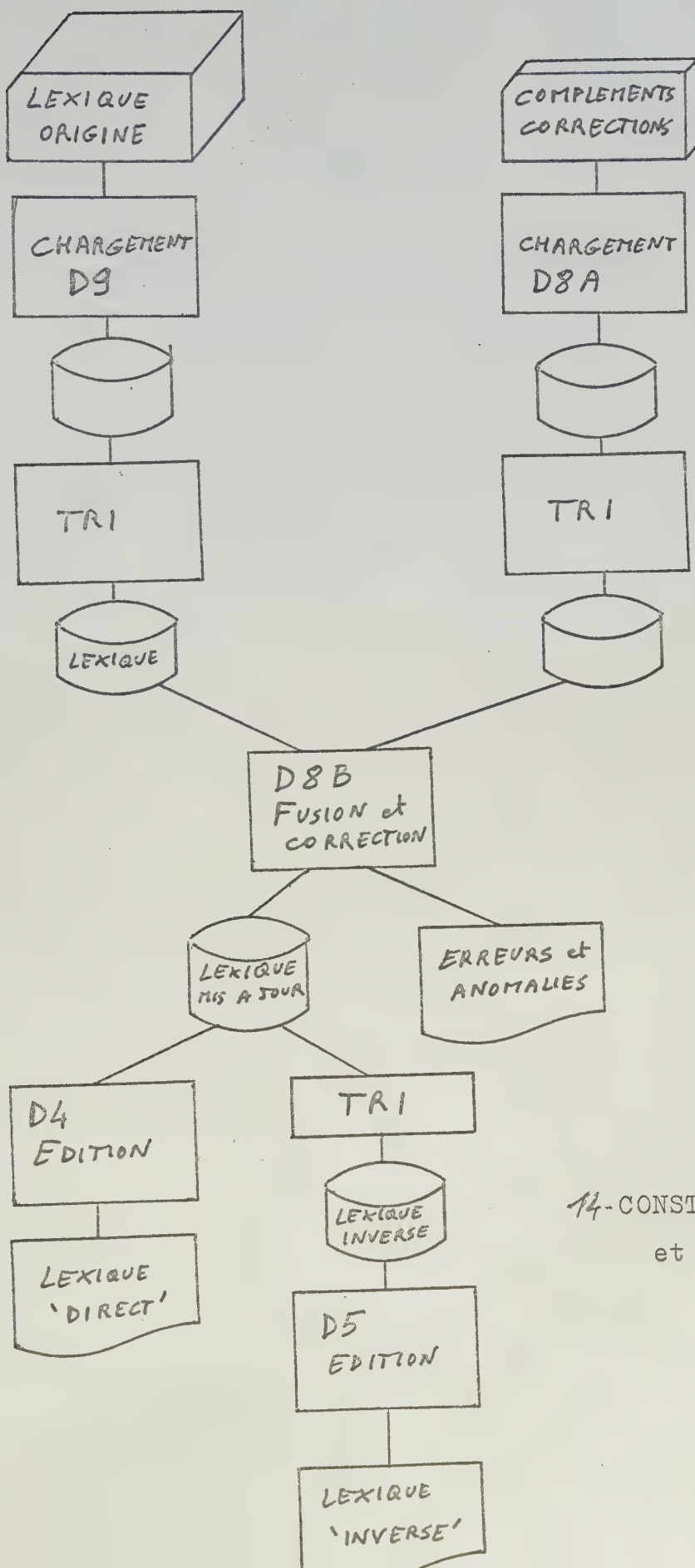
2) - Formes numériques pures. Lors de l'application d'un des programmes de la famille D1, toute chaine de caractères comprise entre deux délimiteurs (espace, barre de fraction ou astérisque) sera extraite comme forme- α . Parmi ces formes- α il en est donc qui sont purement numériques et qui proviennent essentiellement de dates (pas celle du jugement qui n'est pas extraite, mais des dates de lois par exemple) ou de N° d'articles de code. Une solution aurait bien sur consisté à attribuer à chacun de ces nombres un numéro de notion. Mais il faut bien reconnaître qu'il aurait tout de même été bizarre de décider que le nombre 1969 représentait la

notion 13535... Un examen plus précis des textes nous a montré alors que la plupart des formes- α numériques pures étaient constituées de nombres de moins de 5 chiffres (sur les 10 années actuellement traitées, nous n'avons trouvé que deux nombres de plus de 4 chiffres). Il fut donc décidé que toute forme numérique, de 1 à 4 chiffres, serait son propre numéro de notion. Seules les formes de plus de 4 chiffres doivent être mises au lexique avec un N°. Quant aux N° de notions attribués aux formes non numériques, nous les avons fait commencer à 10000.

3) - Les homonymes. Ils représentent le cas le plus embarrassant. Une solution serait d'attribuer alors non pas un mais deux numéros de notion et d'avoir un programme d'analyse automatique qui permette en fonction du contexte de remplacer la forme- α par l'un ou l'autre de ces deux numéros. Ce programme d'analyse risquerait d'être très délicat à mettre au point, d'autant plus que la structure des phrases dans les textes est telle qu'il faudrait étendre l'analyse au-delà de ses limites... En fait, ils sont assez peu nombreux. Nous avons donc pu contourner la difficulté en remplaçant le terme par un synonyme pour l'une des significations ou en altérant son orthographe. C'est ainsi que nous distinguons le DU (û n'existant pas en perforation) de DU qui a le numéro de notion "NUL" en l'écrivant DUS.

En conclusion notre lexique peut donc contenir 10000 notions recouvrant des formes purement numériques (de 0 à 9999) et 9000 notions générales, chacune d'elle étant subdivisible en 10 notions distinctes ce qui donne 90000 notions (de 10000 à 99999). A la fin de 1969, ayant traité tout le vocabulaire de 8 années (1961 à 1968), nous avons un lexique composé de 15000 formes- α (les formes numériques pures ne se trouvant pas physiquement dans le lexique) recouvrant environ 4000 notions. Il semble bien que nous ayons rencontré à présent tout le vocabulaire de base, les formes- α nouvelles étant soit quelques termes rarement utilisés pour une notion déjà connue, soit des termes plutôt techniques et très spécifiques, tels que BUTAGAZ par exemple ...

-o-o-o-o-o-



14-CONSTITUTION, Maintenance
et édition du LEXIQUE

Chapitre V

CREATION ET MAINTENANCE DU LEXIQUE

PROGRAMMES D4, D5, D8, D12

Dans presque tous les cas, l'utilisation du lexique se fait d'une façon séquentielle. Seul le programme de recherche, dans son utilisation en mode conversationnel, nécessite une organisation particulière du lexique. Aussi il est apparu plus simple d'avoir pour le lexique une organisation purement séquentielle. Chaque enregistrement est formé d'une forme- α et du numéro de notion qui lui est associé. Comme dans le cas des fichiers de sortie de D1 et afin de ne pas compliquer inutilement, ces enregistrements sont de longueur fixe, 30 caractères, 5 pour le N° et 25 pour la forme- α .

Le lexique de départ est parforé sur cartes, à raison d'une forme- α par carte avec le N° de notion dans les colonnes 1 à 5 et le mot lui-même cadré à gauche à partir de la colonne 6. Dans ces conditions, la création du premier lexique se ramène à l'utilisation de deux programmes utilitaires standards : un cartes à disque et un tri. Cette façon de procéder ne permet aucun contrôle ce qui oblige à les remettre à plus tard.

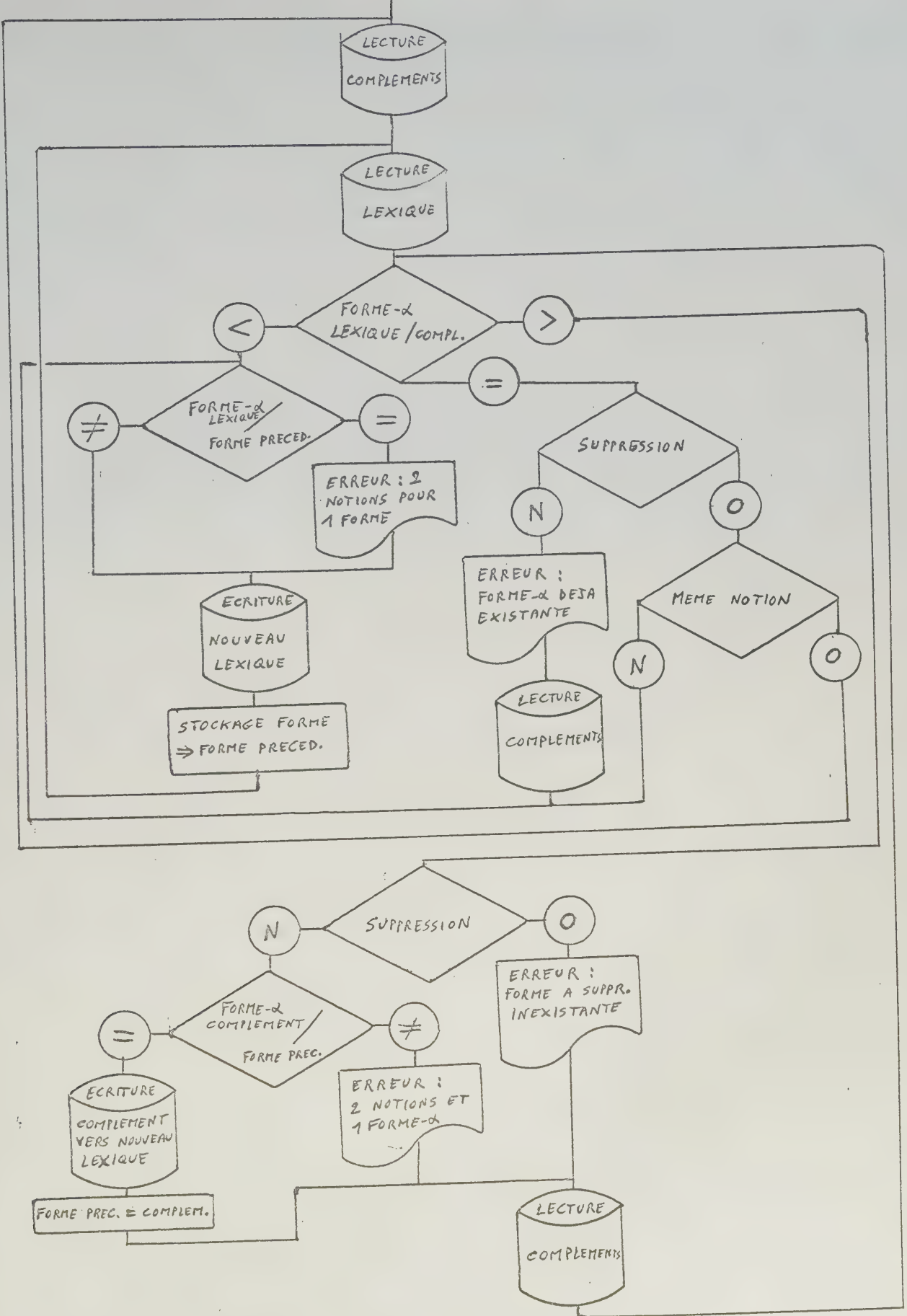


Fig. 11. PRINCIPE DE LA MISE A JOUR DU LEXIQUE.

DOUBLE NUMERO DE NOTION POUR BATTEUSE
CECI N EXISTE PAS ET NE PEUT ETRE DETRUIT 13800 BOVIDE
CETTE FORME EST DEJA CHARGEE, 12090 BOVIN
DOUBLE NUMERO DE NOTION POUR FEVRIER
DOUBLE NUMERO DE NOTION POUR LEGISLATIVE
DOUBLE NUMERO DE NOTION POUR MARS

Figure 11 bis - Message d'erreurs
lors de la mise à jour du lexique.

C'est au cours de l'exploitation du programme D8 servant à la mise à jour du lexique que seront effectués les contrôles. Ils vont consister essentiellement dans la vérification que la même forme- α n'existe pas deux fois dans le lexique. En effet, si le même numéro de notion peut apparaître plusieurs fois avec des formes- α différentes, par contre un mot donné ne peut être associé qu'à une notion unique. Ce programme D8 effectue ce contrôle et sort sur l'imprimante les formes- α présentes dans au moins deux enregistrements.

Ce programme sert en outre à compléter ou modifier le lexique. Pour cela, il accepte en entrée deux fichiers sur disque, le lexique et le fichier des corrections. Ce dernier est obtenu par un processus analogue à la création du premier lexique. La seule différence est que ses enregistrements ont une longueur de 31 caractères, le 31ème étant occupé par un S dans le cas d'un enregistrement à supprimer dans le lexique. Ces deux fichiers, étant triés en ordre croissant des formes- α , sont lus en parallèle, un troisième fichier étant créé en sortie composé de la fusion des deux fichiers exception faite des enregistrements du lexique ayant un correspondant avec S dans le fichier corrections. Là encore, un contrôle est effectué vérifiant que les nouveaux n'existent pas déjà et que les suppressions demandées correspondent bien à quelque chose qui existe (fig. 11 pour l'organigramme de principe et 11 bis pour un exemple de sortie).

40470	DISCULPES	15150	DISCUSSION	15150	DISCUSSIONS	15160	DISCUTANT
15160	DISCUTE	15160	DISCUTE	15160	DISCUTEES	15160	DISCUTER
15160	DISCUTES	45210	DISJONCTEUR	40480	DISJUNCTION	40480	DISJUNCTIONS
30760	DISPARAÎTRE	30750	DISPARITION	30750	DISPARITIONS	30760	DISPARU
30760	DISPARUE	30760	DISPARUES	30760	DISPARUS	15172	DISPENSARE
15172	DISPENSAIRES	15170	DISPENSANT	15170	DISPENSE	15170	DISPENSEE
15170	DISPENSEES	15170	DISPENSER	15170	DISPENSES	26251	DISPONIBILITE
26251	DISPONIBILITES	26250	DISPONIBLE	26250	DISPONIBLES	30770	DISPOSANT
30771	DISPOSANT	30771	DISPOSANTS	30770	DISPOSE	30770	DISPOSEE
30770	DISPOSEES	30770	DISPOSER	30770	DISPOSES	15180	DISPOSITIF
15180	DISPOSITIFS	15190	DISPOSITION	15190	DISPOSITIONS	15200	DISQUE
15200	DISQUES	30780	DISSENTIMENT	30780	DISSENTIMENTS	26260	DISSIMULANT
26260	DISSIMULATION	26260	DISSIMULATIONS	26260	DISSIMULE	26260	DISSIMULEE
26260	DISSIMULEES	26260	DISSIMULER	26260	DISSIMULES	35800	DISSOCIANT
35801	DISSOCIATION	35800	DISSOCIE	35800	DISSOCIEE	35800	DISSOCIEES
35800	DISSOCIER	35800	DISSOCIES	15210	DISSOLUTION	15211	DISSOUTE
15211	DISSOUTES	15220	DISTANCE	15220	DISTANCES	15221	DISSANT
15221	DISTANTES	43512	DISTILLATEUR	43510	DISTILLATION	43511	DISTILLERIE
15241	DISTINCT	15241	DISTINCTE	15241	DISTINCTES	30790	DISTINCTIF
30790	DISTINCTIFS	15240	DISTINCTION	15240	DISTINCTIONS	30790	DISTINCTIVE
30790	DISTINCTIVES	15241	DISTINCTS	15250	DISTRACTION	15250	DISTRACTIONS
15251	DISTRAIRE	15251	DISTRAIT	15251	DISTRAITE	15251	DISTRAITES
15251	DISTRAILS	15251	DISTRAYANT	15261	DISTRIBUANT	15261	DISTRIBUE

Fig. 12. - Un extrait du lexique - Edition directe (D4).

12090

BOVIN

BOVINS

12100

BOXE

BOXEUR

BOXEURS

12110

BOYAUDIER

BOYAUDIERS

12120

BRAS

12130

BRODEUSE

BRODEUSES

12140

BROUILLARD

BRUME

12150

BROUILLON

BROUILLONS

12160

BRUIT

BRUITS

SON

SONS

TAPAGE

Le programme D4 sert à éditer le lexique "direct", c'est-à-dire, classé par ordre alphanumérique croissant (fig. 12). Un tri standard du lexique avec comme argument l'enregistrement entier permet d'obtenir un fichier lexique "inverse", c'est-à-dire, classé par notions. Un tel fichier peut alors être édité par le programme D5, (fig. 13). Le document ainsi fourni permet de retrouver facilement tous les synonymes vrais recouvrant la même notion.

Lors de l'interrogation, comme nous l'avons déjà mentionné, on utilisera les notions, ceci permettant de prendre les localisations de tous les synonymes. Dans les deux premières versions du programme de recherche, nous utilisions effectivement et exclusivement les numéros de notion. Par contre, à partir de la version 3, il est apparu peu pratique d'utiliser cette technique dans le cadre du mode conversationnel. Aussi, dans ce cas, utilise-t-on des formes- α dans la rédaction des questions. Mais comme le fichier inversé est structuré sur les notions, il y a lieu d'effectuer la traduction. Ceci se fait en consultant un lexique à structure de type séquentiel indexé, mais non standard. Nous aurons l'occasion d'expliquer plus loin cette structure et les raisons qui nous poussaient à utiliser une organisation non standard. Ce qu'il nous faut dire ici, c'est qu'il existe un programme particulier, le D12, qui accepte en entrée le lexique organisé en séquentiel et produit en sortie un lexique organisé en vue de son utilisation par le programme de recherche.

Chapitre VI

PREPARATION DU FICHER INVERSE

PROGRAMMES D6A, D6C ET D6D

Parvenus à ce stade, nous possédons deux sortes de fichiers sur disque, ceux produits par la famille D1, composés de formes- α et des localisations associées, et le lexique. Le lexique va permettre le remplacement de la forme- α par le N° de notion et par suite le regroupement des localisations de tous les synonymes.

Si sur un plan théorique, cette seule opération suffit, sur le plan pratique, il y a intérêt à effectuer quelques transformations supplémentaires. En effet, le fichier inversé doit répondre à deux impératifs : être organisé de telle sorte qu'il permette un accès rapide aux localisations de n'importe quelle notion et présenter ces localisations sous une forme condensée permettant un gain de place et de temps. Nous allons examiner en premier ce problème de la condensation.

La condensation va jouer tout d'abord sur la notion elle-même. Dans une première étape, le remplacement de la forme- α par le N° de notion permet de passer de 25 caractères à 5. Mais il est possible de gagner encore de la place en codant ce N° en mode "décimal condensé". Le système 360 admet en ef-

fet plusieurs modes de représentation des données. Parmi ceux-ci, trois sont utilisées dans notre système :

1) - Mode caractère : Quel que soit ce caractère (lettre, chiffre ou signe spécial) il est codé sur un octet (soit une position adressable de mémoire composée de 8 bits).

2) - Mode décimal condensé : Chaque chiffre est codé sur 4 bits ; on a donc deux chiffres par octet, sauf dans le dernier octet de droite qui possède un chiffre et un signe.

3) - Mode binaire : Les nombres sont codés directement en binaire. La mémoire elle-même étant technologiquement binaire, c'est la forme la plus dense. On peut alors prendre des longueurs variant de 1 à 4 octets, suivant la valeur maximale que peut prendre le nombre. Sur un octet, cette valeur est de 255 et elle est de 65535 sur 2 octets.

Le N° de notion pouvant atteindre 99999 est supérieur à 65535 et aurait donc nécessité en binaire 3 octets. Mais, nombre de 5 chiffres, il nécessite également 3 octets en décimal condensé. Nous nous sommes donc contenté de ce dernier mode.

La localisation, sous la forme fournie par D1 est très volumineuse car tous les éléments sont stockés en mode caractère pour simplifier les éditions (en particulier de D3). La partie la plus embarrassante est constituée par la référence de l'arrêt lui-même. Cette dernière, sous la forme année, chambre

et numéro, est peut-être très commode à utiliser manuellement, mais n'est pas du tout adaptée à une utilisation en ordinateur. En effet, on préfère avoir comme référence des numéros en séquence. Or, si on connaît le plus grand numéro dans chacune des chambres de chaque année, il n'y a aucune difficulté à établir une table de correspondance.

Afin d'alléger les programmes qui auront à utiliser cette table elle est fabriquée par un programme particulier (VDAR) et stockée sur disque. Les renseignements nécessaires lui sont fournis sur cartes dans l'ordre année-chambre. Le numéro 1 étant attribué au premier arrêt de la première chambre de la première année chargée, la table est constituée d'une succession de groupes de 5 octets : année (2), chambre (1) et un nombre en binaire sur 2 octets qui est le numéro "séquentiel" du dernier arrêt de la chambre précédente.

L'utilisation de cette table permet donc de remplacer la référence de départ par un nombre codé en binaire sur deux octets. On voit donc apparaître ici une limitation due à l'utilisation de 2 octets seulement. On ne peut pas traiter dans le même fichier plus de 65535 arrêts. En fait, cette limite ne tient pas à la méthode utilisée mais simplement à la longueur attribuée. Le principe reste valable en utilisant 3 octets ... Mais comme nous ne pensions pas avoir à dépasser 40000 arrêts, nous avons plutôt intérêt à économiser de la place. Nous devons enfin signaler que l'utilisation de cette

référence "séquentielle" est purement interne, c'est-à-dire, que l'utilisateur n'aura jamais l'occasion de la connaître. En effet, si elle est utilisée dans le fichier inversé, le programme de recherche va utiliser la même table de conversion avant d'éditer les réponses qui apparaîtront donc avec les références habituelles.

La condensation du reste de la localisation n'offre plus aucune difficulté. Le numéro de paragraphe est gardé en mode caractère sur un octet (il l'est déjà et d'autre part on se réserve ainsi la possibilité d'utiliser des lettres au cas où il y aurait plus de 9 paragraphes). Par contre, le numéro de phrase est transformé en binaire sur un octet (maximum de 255 phrases dans un paragraphe) ainsi que le N° de mot dans la phrase.

Ainsi , en appliquant les techniques énoncées ci-dessus, chaque enregistrement des fichiers produits par un des programmes de la famille D1 donnera naissance à un nouvel enregistrement de longueur 8 octets. Il est composé de 3 octets pour la notion, 2 pour la référence d'arrêt, 1 pour le paragraphe, 1 pour la phrase et 1 pour la place du mot. C'est le programme D6A qui effectue ce travail. A vrai dire, de même qu'il existe deux versions de D1, D1 et D1B, il en existe deux de D6A, D6A acceptant un seul fichier en entrée et D6AB qui en accepte deux en entrée pour n'en faire qu'un seul en sortie. C'est ce dernier qui est habituellement utilisé.

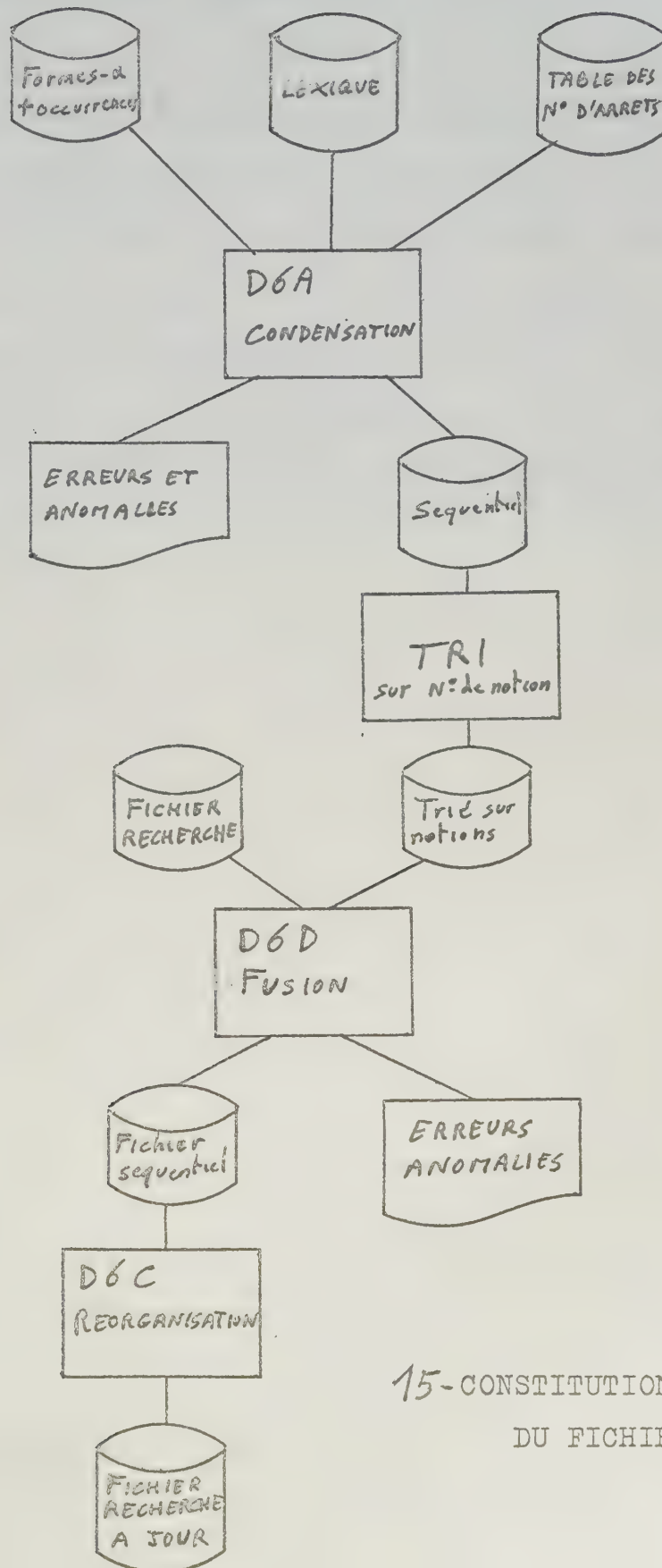
Ce programme utilise également le lexique pour remplacer les formes- α par les numéros de notion. Au cours de cette opération trois cas particuliers peuvent se présenter:

1) - La forme est numérique pure : Si son nombre de chiffres est supérieur à 4, elle est traitée comme une forme non numérique. Dans le cas contraire, ce nombre est tout simplement transformé en décimal condensé.

2) - La forme- α n'existe pas au lexique : Elle est alors listée sur l'imprimante accompagnée de sa localisation. C'est ce qui explique que ce programme soit utilisé en vue d'effectuer les corrections sur le fichier d'origine. Rien n'est placé sur le fichier de sortie.

3) - Le numéro de notion correspondant est "NUL" : On passe à l'enregistrement suivant du fichier d'entrée sans rien écrire sur celui de sortie.

Le travail qui suit est un tri standard du fichier de sortie en utilisant comme argument de tri l'enregistrement entier. On obtient alors à la suite toutes les localisations pour la même notion, elles-mêmes classées par ordre croissant. S'il existe déjà un fichier de recherche (fichier inversé) on doit alors le fusionner avec celui qu'on vient d'obtenir. Nous aurions pu prévoir une structure telle du fichier de recherche qu'il soit possible de le compléter. Mais notre optique consiste à mettre à jour ce fichier par gros morceaux, en général par année entière. Dans ces conditions il est plus intéressant de



15-CONSTITUTION ET MAINTENANCE
DU FICHIER DE RECHERCHE

réorganiser le fichier complètement. C'est ce que nous faisons (fig. 15).

Pour cela le programme D6D, acceptant en entrée le fichier de recherche déjà existant et le lisant d'une façon séquentielle et le fichier obtenu par D6A et trié, fabrique en sortie un fichier analogue à celui de D6A, classé par notion et à l'intérieur par localisations. Il ne reste plus qu'à réorganiser le fichier de recherche comme il va être expliqué dans le chapitre suivant. C'est ce que fait le programme D6C.

-o-o-o-o-

Chapitre VII

ORGANISATION DU FICHIER INVERSE

Un fichier inversé servant à la recherche doit être organisé d'une façon telle qu'on ait la possibilité d'accéder de la façon la plus rapide possible à n'importe laquelle des notions qu'il contient. Il est donc évident de suite qu'une organisation simplement séquentielle ne saurait convenir...

Notre système est prévu pour tourner sous le contrôle du Disk Operating System d'IBM . Ce dernier offre trois organisations de fichier :

1) - Séquentiel : Le fichier ainsi organisé ne peut être lu ou écrit que d'une façon séquentielle, les différents enregistrements ne pouvant être traités que dans un ordre immuable.

2) - Séquentiel indexé : Dans cette organisation, chaque enregistrement est muni d'une clef, l'ensemble du fichier étant classé dans l'ordre croissant de ces clefs. Le fichier est d'autre part découpé en deux parties, un "index des cylindres" et les données. L'index des cylindres va comporter les numéros de cylindre de la partie des données, chacun étant associé à la clef la plus haute contenue dans ce cylindre. Une lecture de cet index (rapide car peu volumineux) permet donc de retrouver sur

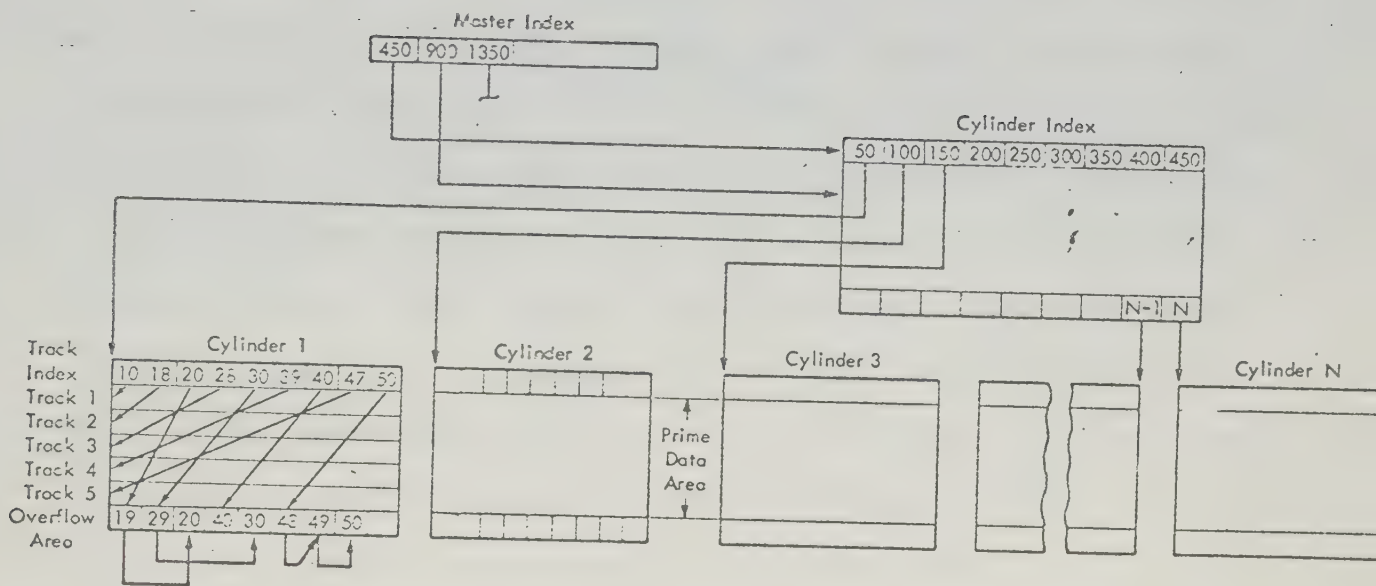


Figure 16 - Organisation générale d'un fichier en Séquentiel Indexé.

quel cylindre doit se trouver la clef cherchée dans la mesure où elle existe.

En tête de chaque cylindre de la partie des données, on trouve encore un index, celui de pistes. Analogue à celui des cylindres, il permet de trouver sur quelle piste se trouve l'enregistrement correspondant à la clef cherchée. Il faudra donc au minimum trois lectures (mais un seul déplacement des têtes de lecture, les deux dernières lectures s'effectuant sur le même cylindre) pour atteindre les données. De plus cette organisation permet des adjonctions sans réorganisation complète grâce à des "zones d'overflow" dans lesquelles les enregistrements sont "chainés" les uns aux autres (fig. 16). Mais cette organisation présente une contrainte, les enregistrements doivent être de longueur fixe et on ne peut pas avoir plusieurs enregistrements munis de la même clef.

3) - Accés direct : chaque enregistrement sur un fichier disque est muni d'une adresse. Cette adresse est composée du N° de cylindre, du N° de piste et enfin du N° de l'enregistrement sur la piste. Si on connaît cette adresse, on peut dans une telle organisation aller lire ou écrire directement l'enregistrement. Il est bien évident que c'est cette méthode qui offre la plus grande vitesse d'accès puisqu'il n'y a plus qu'un seul accès. Cette méthode pourra être utilisée chaque fois qu'on possèdera une méthode permettant de calculer l'adresse disque à partir d'une référence quelconque.

Nous allons voir à présent ce que ces trois méthodes nous offraient comme avantages et inconvénients.

1) - Séquentiel : Un tel fichier est pratiquement inutilisable sauf éventuellement pour des essais très réduits portant sur un très petit fichier.

2) - Séquentiel indexé : Son principe est très séduisant puisqu'il ne suppose connue que la clef qui pourrait parfaitement être le N° de notion. Le seul inconvénient, mais de taille, vient du fait qu'il n'accepte que des enregistrements de longueur fixe. Or, il est bien évident que l'information associée à chaque clef, c'est-à-dire, l'ensemble de toutes les localisations de cette notion, est de longueur parfaitement variable dans des limites difficiles à préciser. On peut toutefois estimer d'après des sondages effectués dans notre fichier, que certaines notions peuvent n'avoir qu'une localisation alors que d'autres, sur dix années, atteignent les 5000. Nous avons bien envisagé de regrouper les localisations par groupes (de 10 par exemple) chaque groupe ayant une clef composée de la notion suivie d'un N° de séquence du groupe à l'intérieur de cette notion. L'inconvénient aurait alors été qu'il aurait fallu autant d'accès sur disque que de groupes pour une notion et que la diminution du nombre d'accès par augmentation de la taille des groupes nous aurait exposé à avoir un très mauvais taux de remplissage ...

3) - Accès direct : La grande diversité de longueur de l'information associée à chaque notion rend très difficile, sinon impossible, le calcul d'une adresse disque à partir du N° de notion. Ce calcul est par contre possible si on donne une longueur fixe pour chaque notion. On pourrait alors envisager d'avoir deux fichiers. Le premier avec un enregistrement par notion, de longueur fixe, donnerait comme information l'adresse disque de l'enregistrement de l'autre fichier dans lequel on trouverait les localisations de cette notion. Il y a tout de même un inconvénient, c'est d'avoir 99999 notions, ce qui oblige à prévoir autant d'enregistrements dans le premier fichier. Dans ce cas, il est alors nécessaire de prévoir plusieurs cylindres, avec un taux de remplissage faible et une réorganisation des deux fichiers en parallèle à chaque mise à jour...

La conclusion est bien qu'aucune des trois méthodes proposées par IBM n'est parfaitement adaptée à notre cas. Il ne nous restait donc plus qu'à bâtir notre propre organisation. Ceci est réalisable grâce à une méthode d'accès commune à toutes les unités périphériques, dite accès physique. Il nous faut ici expliquer succinctement comment s'effectuent les opérations d'entrée-sortie sur le 360.

Elles sont effectuées par des canaux, ensemble de circuits logiquement distincts de l'unité centrale, ce qui permet une certaine simultanéité des opérations d'entrée-sortie et de calcul. Le travail de l'unité centrale se réduit à déclencher

le départ d'un programme de canal en spécifiant l'unité concernée et l'adresse de la mémoire centrale où se trouve le programme de canal. Ce dernier est constitué par une suite d'instructions particulières appelées CCW (Channel Command Word).

Lors de la terminaison de l'opération, ou en cas d'incident dans son déroulement, le canal interrompt le fonctionnement de l'unité centrale qui est ainsi tenue au courant de ce qui se passe. Dans le cadre d'un fonctionnement sous contrôle d'un système d'exploitation, et en particulier dans l'optique d'un fonctionnement en multi-programmation, il est indispensable que le superviseur ait la haute main sur la gestion des périphériques. Aussi c'est lui et lui seul qui peut utiliser l'instruction de départ de programme canal.

Mais les méthodes d'accès standard, dites logiques, ne sont pas incluses dans le superviseur et forment des "sections de contrôle" particulières incluses dans le programme. Ces dernières provoquent les entrées-sorties par un appel au superviseur après avoir placé en un endroit convenu l'adresse d'un bloc d'information appelé CCB (Channel Control Bloc). Dans ce dernier, on trouve l'indication de l'unité concernée, l'adresse du programme canal, etc... Il est donc parfaitement possible à un programme quelconque de fabriquer un tel CCB, de préparer les CCW d'un programme canal et enfin de faire un appel au superviseur pour lui faire déclencher l'opération... C'est cette technique qui est appelée accès physique.

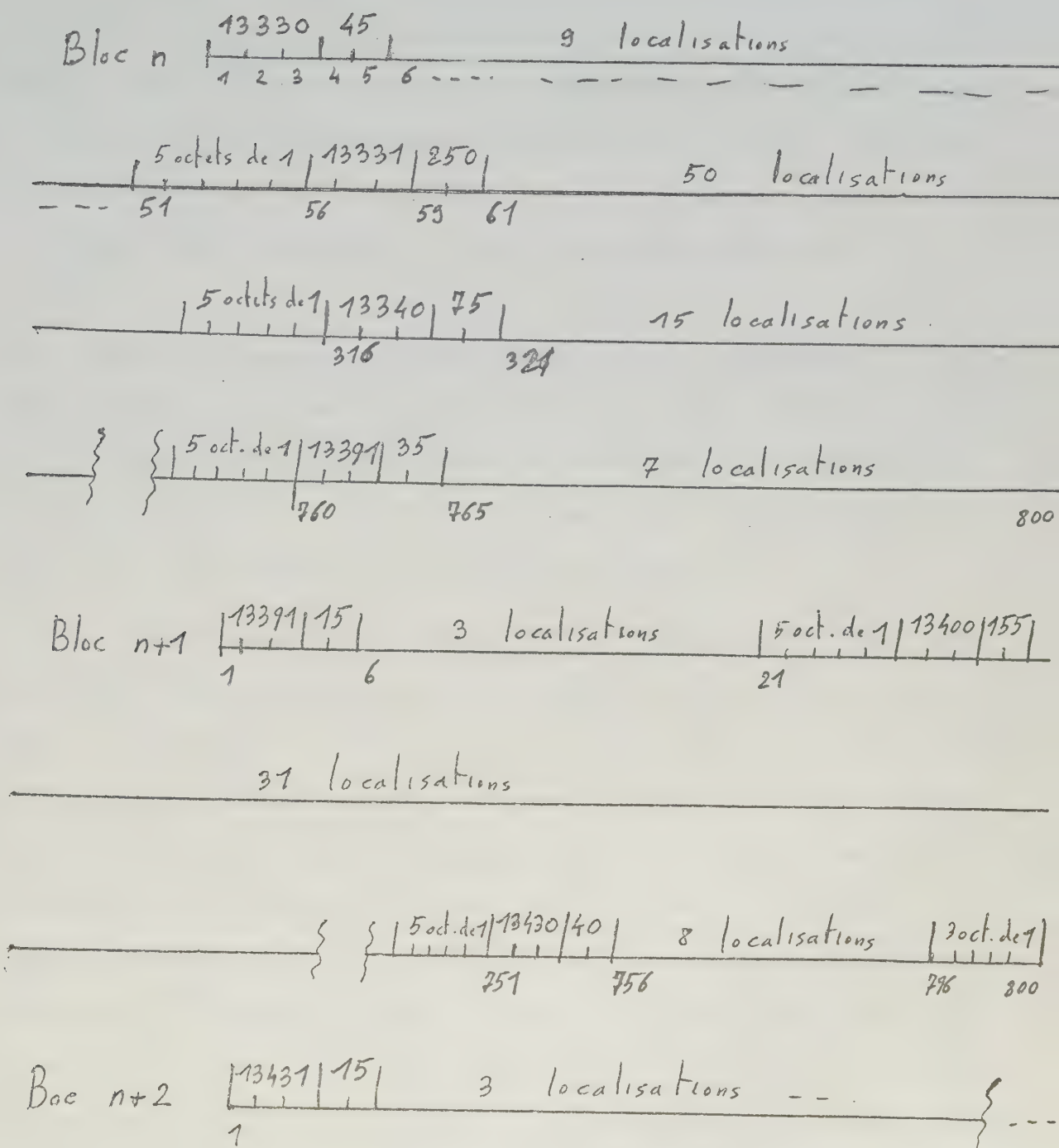
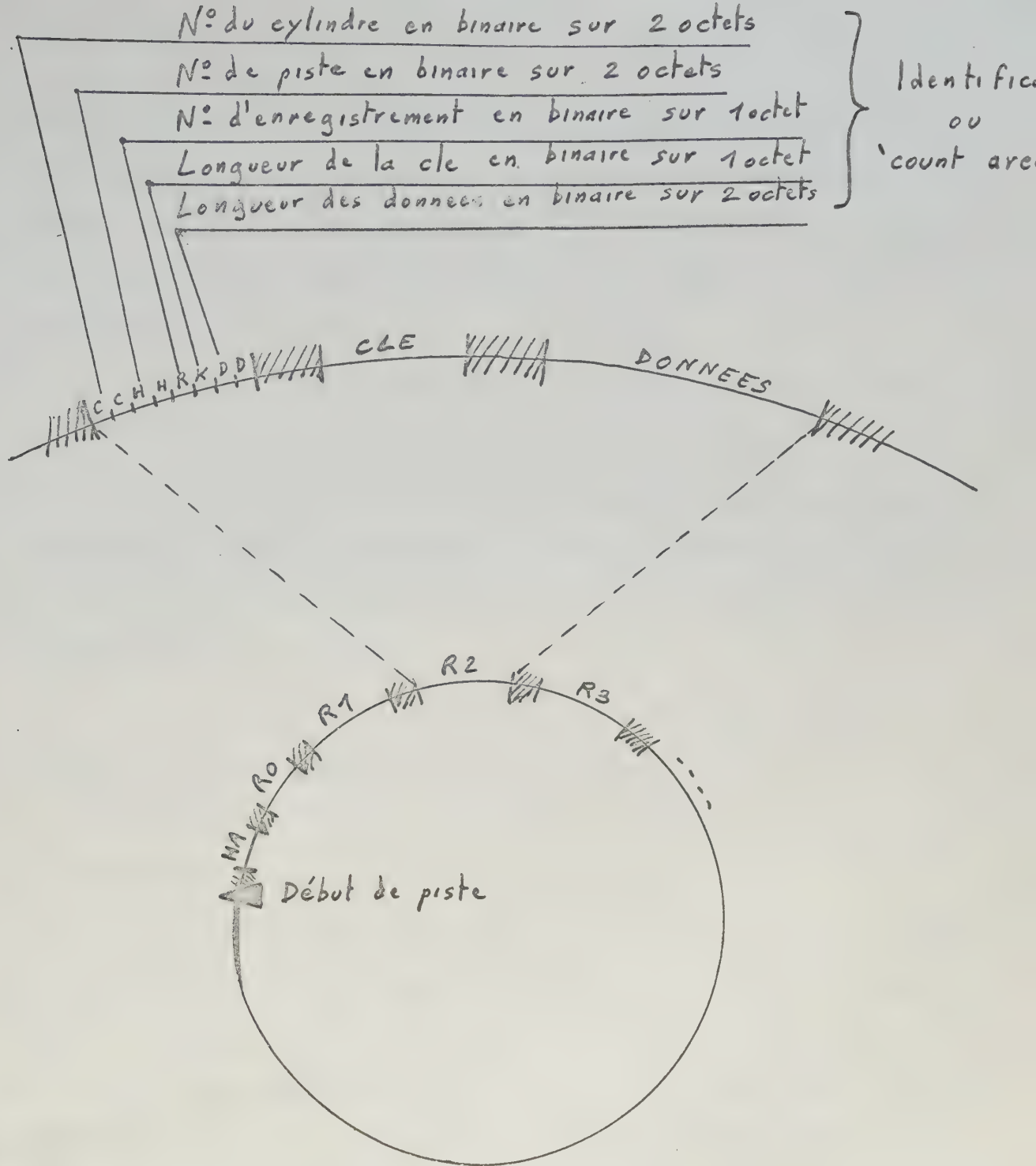


Fig. 17. Structure des enregistrements du fichier de recherche. On remarquera la notion 13391, avec 10 localisations, à cheval sur deux blocs.

Une telle façon de procéder oblige à écrire entièrement les programmes de canal, et par suite à connaître les instructions propres à ce type de programme. Cela suppose aussi une connaissance minimum de la technologie ou tout au moins du fonctionnement des périphériques. Mais l'énorme avantage est de pouvoir faire exécuter à l'unité périphérique toutes les fonctions qui ont été prévues par le constructeur et, sur disque en particulier, de pouvoir utiliser des structures non standard adaptées à tel ou tel problème particulier. En fin de compte, la conclusion que nous avons pu tirer de cette expérience et des autres qui ont suivi, c'est que cette méthode n'est finalement pas très difficile à mettre en oeuvre.

Notre fichier de recherche va donc pouvoir être organisé d'une façon particulière, en fonction de nos besoins propres. La rapidité d'accès nous a fait opté pour une organisation proche du séquentiel indexé, mais sans zone d'overflow (notre mise à jour devant s'effectuer par adjonctions importantes, une réorganisation s'impose à chaque fois si on veut conserver de hautes performances). Mais il doit pouvoir présenter de l'information de longueur variable pour chaque clef. Qui plus est, cette longueur peut être telle qu'il soit nécessaire d'utiliser plusieurs pistes, voir plusieurs cylindres pour une seule notion..

Les données sont formées de blocs de 800 octets et il y a donc quatre blocs par piste. A l'intérieur de chaque bloc (fig. 17) l'information relative à une notion commence par le N° de notion lui-même en décimal condensé sur 3 octets. Il est sui-



HA = 'Home Address', bloc d'information donnant l'adresse de la piste (N° de cylindre et de piste).

RO = Enregistrement 0, contient des informations sur l'état de la piste.

R1, R2, ..., Rn = Enregistrements ordinaires accessibles au programme, de longueur variable et en nombre variable.

Fig. 18. STRUCTURE D'UNE PISTE DE DISQUE MAGNETIQUE.

vi de la longueur occupée par les localisations dans ce bloc, cette longueur étant exprimée en binaire sur 2 octets. Les localisations suivent alors à raison de 5 octets par localisation (le N° de notion n'étant plus répété). Si c'est la fin des localisations de cette notion elles sont suivies de 5 octets remplis de 1 binaires, ce qui ne peut être confondu avec aucune localisation. Une telle structure présente deux avantages :

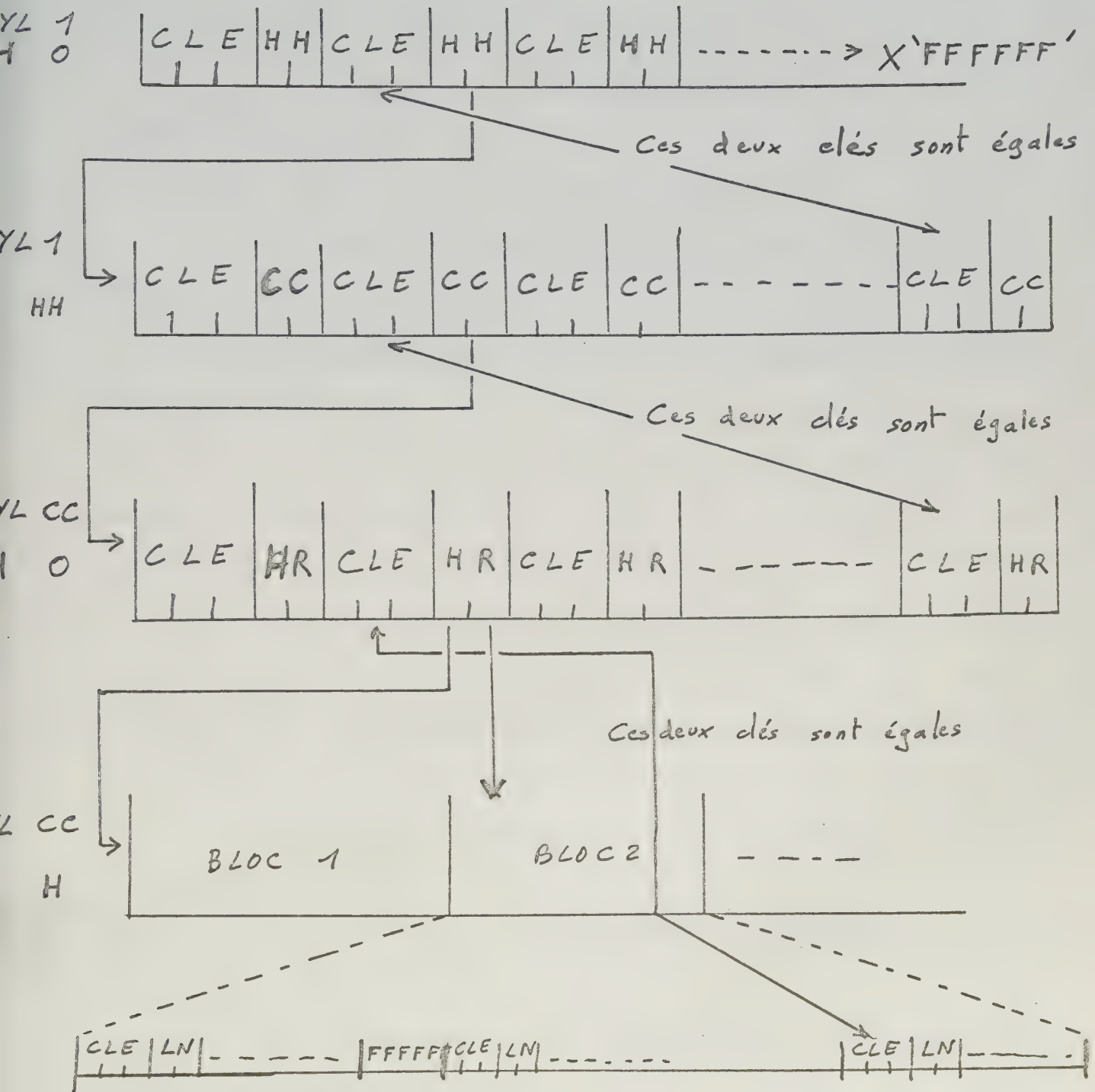
1) - Lorsqu'un bloc a été transféré dans la mémoire centrale, on atteint rapidement la notion cherchée en sautant par dessus celles qui la précèdent grâce à la connaissance de la longueur qu'elles occupent.

2) - La vérification de la présence des 5 octets de 1 derrière les localisations de chaque notion permet de vérifier la structure et de détecter toute destruction accidentelle de l'information.

Si par contre les localisations de cette notion débordent de ce bloc, elles occuperont toute la fin du bloc, sans avoir 5 octets de 1, les localisations continuant sur le bloc suivant en tête duquel sera répété le N° de notion (d'où vérification possible) suivi de la longueur de ces localisations sur ce nouveau bloc.

Il nous faut indiquer à présent comment est organisée l'information sur disque, d'une façon générale (fig. 18). Tout enregistrement possède au moins deux parties :

1) - L'identification d'une longueur 8 octets, compre-



19- ORGANISATION DES FICHIERS DE RECHERCHE
ET DES TEXTES.

nant l'adresse disque suivi de la longueur de la clef en binaire sur un octet et de la longueur des données, également en binaire mais sur deux octets.

2) - Les données : La troisième partie, qui est optionnelle, est formée de la clef. Si elle est présente, elle est placée entre l'identification et les données.

Les blocs d'information contenant les localisations ne possèdent pas de clef.

Les instructions composant les programmes de canal pour les disques sont de quatre sortes :

1) - SEEK qui permet le positionnement mécanique des bras de lecture sur un cylindre particulier ainsi que la commutation des têtes sur la bonne piste.

2) - SEARCH qui sont des branchements conditionnels sur une condition portant soit sur les identifications, soit sur les clefs, par comparaison entre ce qui est lu sur le disque et ce qui se trouve dans la mémoire centrale à une certaine adresse. Si la condition est remplie, le programme saute le CCW suivant.

3) - TIC qui est un branchement inconditionnel au CCW situé à une certaine adresse.

4) - READ ou WRITE qui lit ou écrit les données du premier endroit qui se présente.

Pour accéder aux localisations d'une notion donnée, nous allons utiliser trois index (fig. 19).

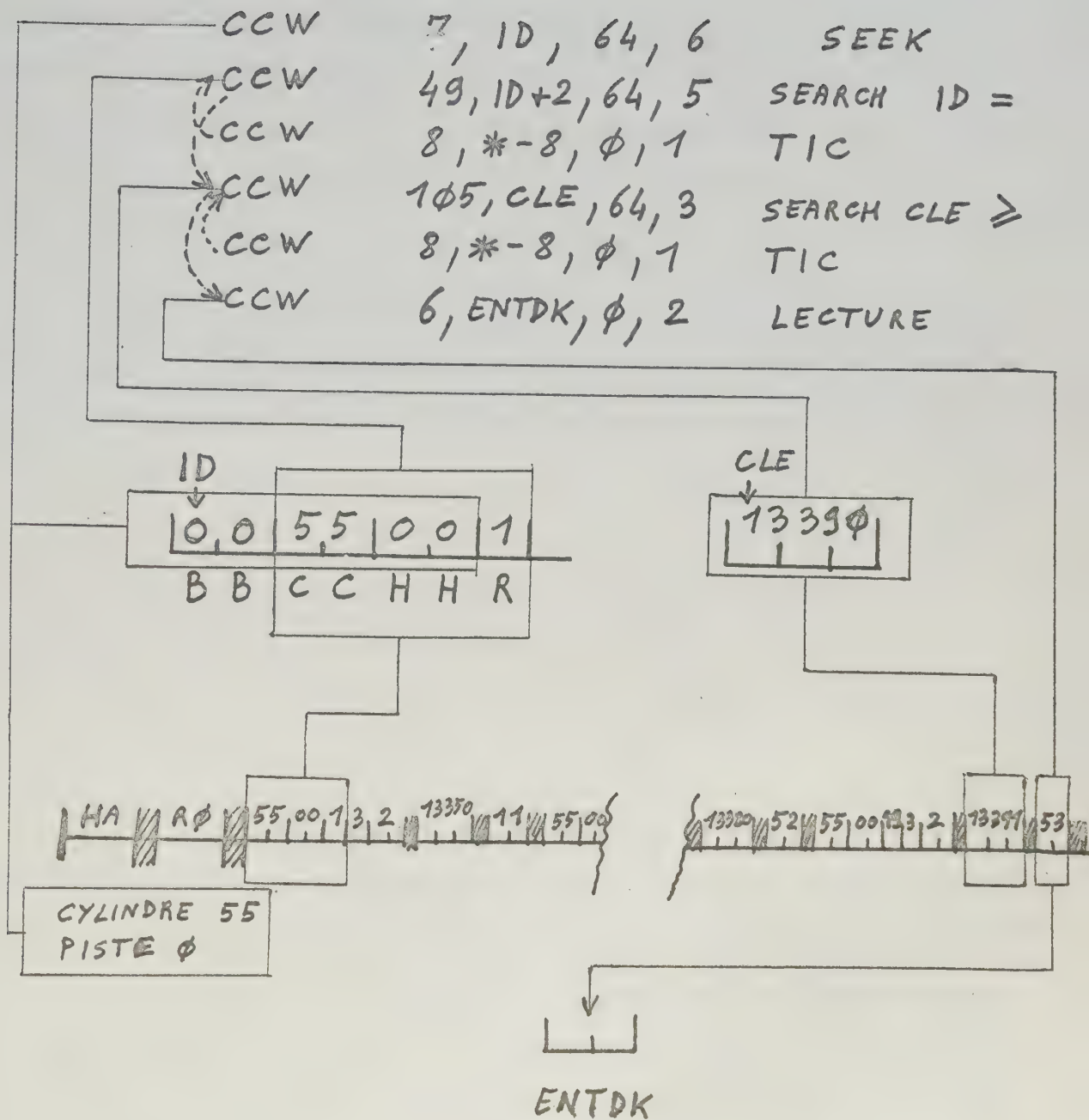


Fig. 20 - Programme canal de lecture d'index.

1) - L'index principal chargé en mémoire centrale au début du traitement et stocké entre temps sous forme d'un enregistrement unique sur la première piste du premier cylindre du fichier. Il possède jusqu'à 9 entrées, une pour chacune des pistes restantes de ce premier cylindre, chaque entrée étant composée d'une notion correspondant à la plus haute clef sur chaque piste suivie du N° de piste.

2) - L'exploitation de ce premier index permet de trouver sur quelle piste de l'index de cylindre il y a lieu de chercher. Cet index est composé d'enregistrements complets avec les trois parties, identification, clef et données. Il y a un enregistrement par cylindre du fichier, la clef étant la plus haute notion contenue dans ce cylindre et la donnée correspondant au N° du cylindre. Par une recherche sur clef égale ou supérieure, on peut à l'aide d'un seul accès sur disque trouver le cylindre cherché.

3) - En tête de chaque cylindre, occupant toute la première piste, se trouve le dernier index. Celui-ci possède un enregistrement par bloc de données dans le cylindre, soit 36 enregistrements. Chaque enregistrement possède une clef égale à la plus haute notion contenue dans le bloc et comme données, le N° de piste et le N° d'enregistrement sur la piste. De cette façon on peut accéder au bloc directement par son identification.

Pour ne pas alourdir cette description, nous avons détaillé sur la figure 20 le programme canal permettant l'explo-

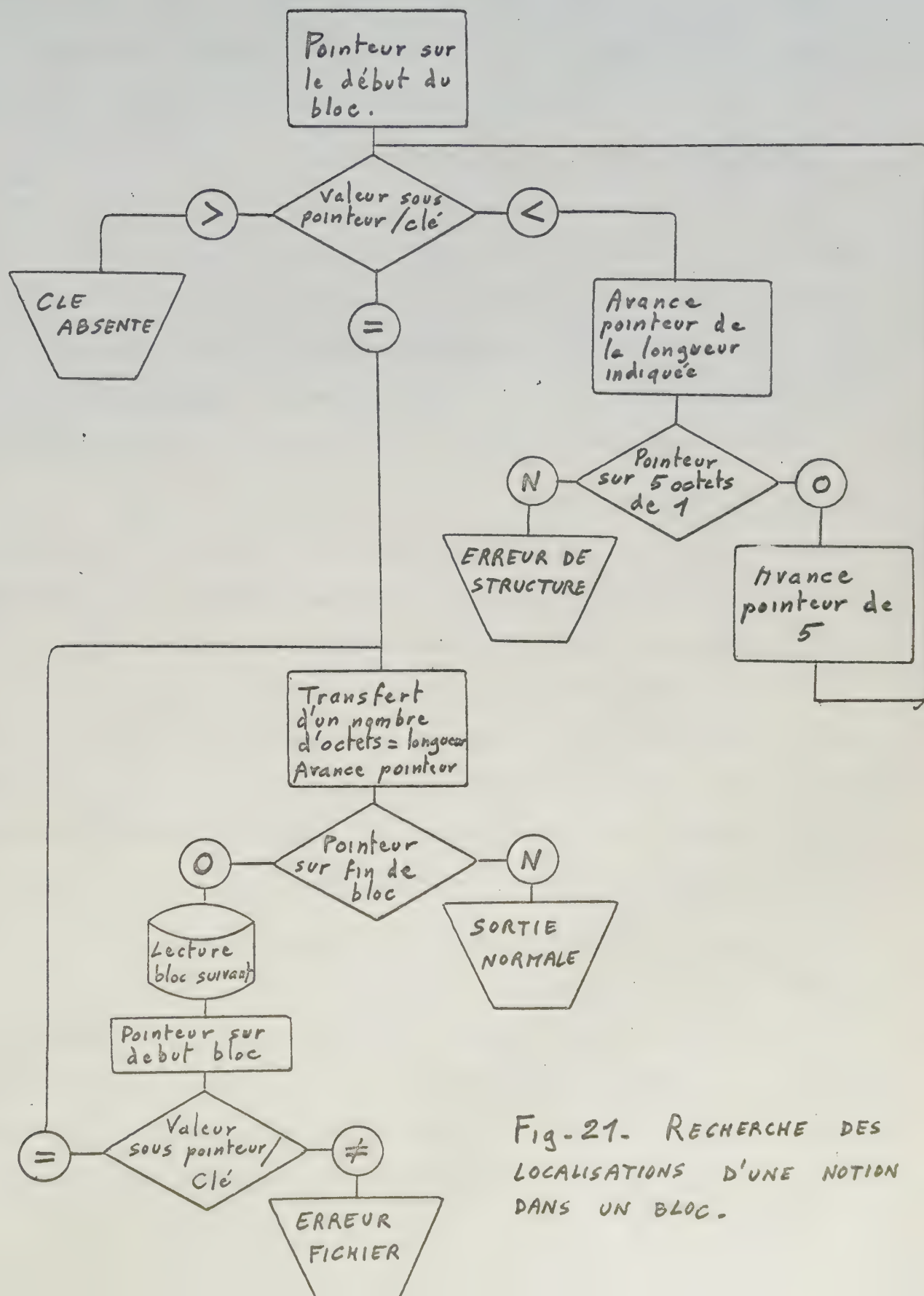


Fig-21. RECHERCHE DES LOCALISATIONS D'UNE NOTION DANS UN BLOC.

tation des 2ème et 3ème index et sur la figure 21 , le détail des opérations permettant de retrouver le premier bloc contenant les localisations d'une notion donnée.

Ce qu'il y a lieu de remarquer, c'est qu'il n'y a aucun inconvénient à ce que plusieurs blocs soient référencés par la même notion. En effet, même si des clefs sont identiques, le fait de balayer les index en partant du début de la piste (ce qui est obtenu par un premier SEARCH sur l'identification du premier enregistrement) avec un SEARCH sur clef égale ou supérieure donnera toujours le premier bloc référencé par cette notion.

Si les localisations d'une notion nécessitent plusieurs blocs, il n'y a aucune difficulté à obtenir les blocs suivants. En effets, on connaît exactement l'identification du premier bloc, c'est-à-dire, son adresse disque exacte. Sachant qu'il y a 4 blocs par piste (N° d'enregistrement sur la piste de 1 à 4), que les blocs se trouvent sur les pistes 1 à 9 de chaque cylindre (la piste 0 étant occupée par le dernier index) on peut sans peine en déduire l'adresse disque exacte du bloc suivant. Il n'y a plus alors qu'à le lire directement par son identification.

Que pouvons-nous dire en conclusion de ce chapitre ? C'est que nous possédons maintenant une méthode d'accès efficace qui ne nous a pas causé plus d'ennuis que n'importe laquelle des méthodes standard . Les sous-programmes nécessaires à son exploitation étant très spécifiques sont en fin de compte moins volumineux que les sections de contrôle d'accès standard. Enfin

notre méthode est très compétitive, tant sur le plan du taux de remplissage des fichiers que sur celui de la vitesse d'accès. Nous en arrivons même à penser que l'utilisation de telles méthodes d'accès physique gagneraient à être un peu plus généralisées... Mais comme ce sujet touche celui combien brulant de savoir s'il vaut mieux utiliser des langages évolués, faciles mais rigides, ou des langages plus élémentaires et plus proches de l'ordinateur, nous préfèrerons en rester là...

-o-o-o-o-o-

Chapitre VIII

LE FICHER DES TEXTES

PROGRAMMES D10, D10B, D10T ET D10Q

Pour trouver les réponses à une question, le seul fichier inversé suffit. Mais si on se limite à lui, on ne pourra exprimer la solution que sous la forme des références des arrêts.

Nous avons pensé tout d'abord que cette façon de répondre pouvait parfaitement suffire. En effet, la concision extrême des documents de base ne permet pas une interrogation tellement fine qu'on puisse être certain que la solution donnée corresponde exactement à ce qu'on cherchait. Du reste est-ce bien souhaitable ? Ce n'est pas du tout certain. En effet, si on cherche dans le fichier c'est pour voir s'il n'y existe pas déjà une affaire présentant quelque ressemblance avec un cas précis. Mais il est bien évident qu'il y a peu de chance d'y trouver un arrêt rendu dans des circonstances exactement semblables. Si on interrogeait avec trop de précision, on risquerait tout simplement d'ignorer certains arrêts non dépourvus d'intérêt...

Ayant donc interrogé d'une façon plutôt large, il faut s'attendre à avoir un certain "bruit", c'est-à-dire, un

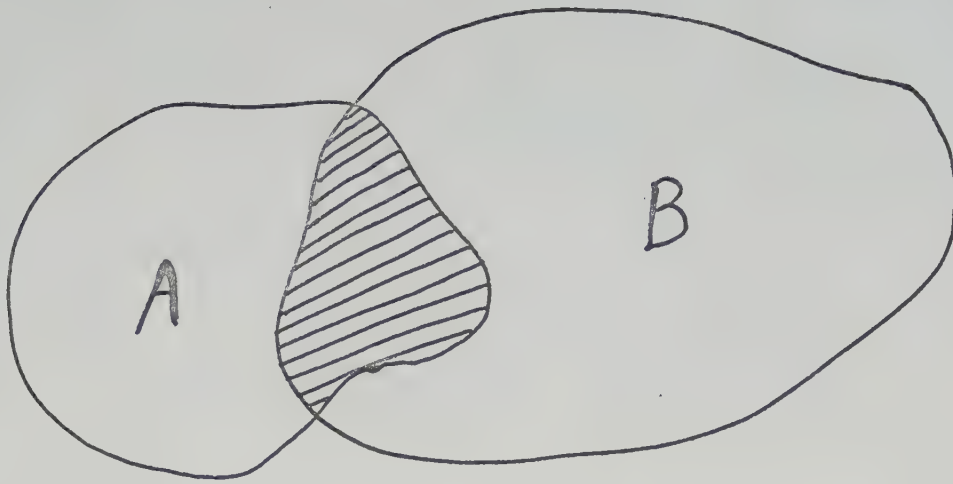
certain nombre de reponses parasites. Ce n'est normalement qu'à la lecture du texte complet de l'arrêt qu'on pourra se rendre finalement compte de l'intérêt qu'il peut présenter. Seulement l'édition du Bulletin des arrêts de la Cour de Cassation, rien que pour les chambres civiles, représente un volume par mois. Si on a un certain nombre de réponses dispersées sur l'ensemble de dix années , c'est à la limite les 120 volumes qu'il va falloir consulter.

Il nous est alors apparu que si en dernier ressort, seule la lecture de l'arrêt complet permet de juger de l'intérêt, la seule lecture du titre principal (c'est-à-dire, de ce qui est pour nous le document de base) permet d'éliminer un certain nombre de références comme appartenant à des arrêts manifestement sans intérêt. Il devenait donc utile, sinon indispensable, de pouvoir éditer les solutions d'une question sous forme des titres principaux des arrêts sélectionnés. Comme dans le cas du fichier inversé, on a le cas d'une information de longueur variable associée à chaque référence (quoique dans des proportions moindres). Nous avons donc utilisé la même solution, la clef étant cette fois un nombre binaire sur 3 octets obtenu par conversion directe de la référence externe (année, chambre et N°) considérée comme un nombre de 7 chiffres. L'information est simplement constituée des images des cartes composant le fichier de départ.

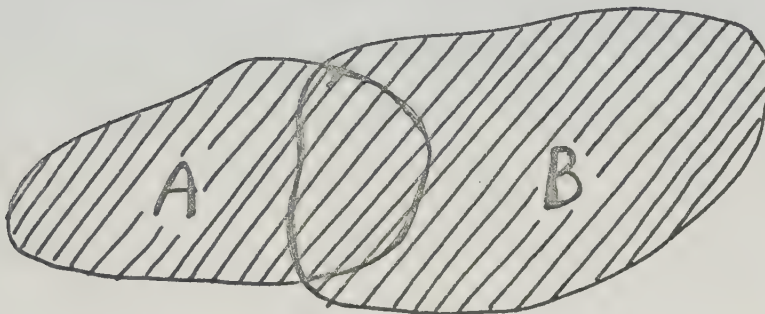
La constitution de ce fichier est obtenue par un des programmes de la famille D10.

D10 permet de créer le début du fichier à partir des cartes. D10B permet de compléter le fichier également à partir des cartes. Nous avons admis (tout au moins pour le moment, cette partie n'étant pas considérée comme essentielle) que la complémentation de ce fichier s'effectuait toujours avec des arrêts chronologiquement postérieurs à ceux déjà chargés. Le programme commence donc par rechercher la limite atteinte dans les différents index et les ayant trouvées, poursuit le chargement. D10T est analogue à D10 mais accepte en entrée le fichier séquentiel indexé des cartes-arrêts créé par le programme D1T. Enfin, D10Q est semblable à D10B mais en utilisant la même source que D10T.

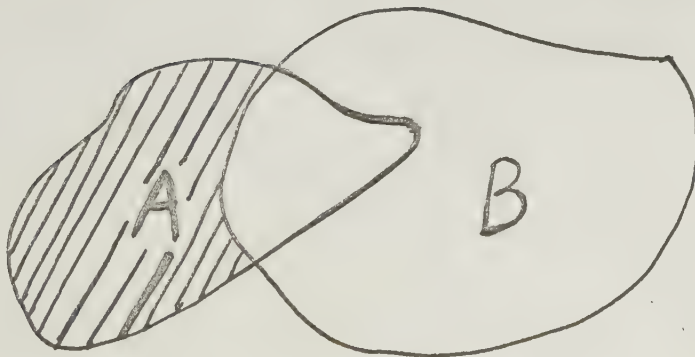
-O-O-O-O-O-



A ET B



A OU B



A SAUF B

Chapitre IX

LES PRINCIPES DE LA RECHERCHE

ET LA STRUCTURE DES QUESTIONS

En simplifiant à l'extrême, on peut dire que le premier principe de toute recherche est de trouver quels sont tous les textes dans lesquels se trouve telle notion. Il est bien évident que la simple consultation du fichier inversé fourni la solution. En pratique, il sera très rare de se contenter de cela. Il arrive pourtant parfois que ce soit le cas, en particulier dans des recherches mettant en jeu des notions très précises et peu courantes. C'est ainsi que lors d'une interrogation sur une histoire de responsabilité dans un accident survenu à un spectateur lors d'une corrida, il a suffi d'interroger sur taureau, les arrêts rendus en cette matière étant particulièrement rares à la Cour de Cassation.

Le plus souvent l'interrogation se fera par l'intermédiaire d'une combinaison de notions à l'aide d'opérateurs logiques. Les trois opérateurs utilisés sont ET, OU et SAUF (figure 22). Si on note (A) l'ensemble des arrêts dans lesquels on trouve la notion A, (B) celui de ceux contenant B et (S) l'ensemble des arrêts solution, on aura :

Pour A et B, $(S) = (A) \cap (B)$, c'est-à-dire, qu'on au-

ra tous les arrêts dans lesquels se trouvent simultanément A et B.

Pour A ou B, $(S) = (A) \cup (B)$, c'est-à-dire tous les arrêts contenant soit A, soit B, soit les deux.

Pour A SAUF B, $(S) = (A) \cap C(B)$, c'est-à-dire, ceux dans lesquels on trouve A mais pas B.

Bien entendu on peut exprimer les conditions à l'aide de plus de deux notions. Dans ce cas l'ordre dans lequel est effectué la recherche de la solution peut être soit simplement celui d'apparition des notions, soit régi par des priorités relatives des opérateurs, soit encore modifié par l'utilisation de parenthèses. On trouvera sur la figure 23 la représentation graphique des différentes solutions à une même question suivant les cas.

Si on se limite à ce qui précède, on ne va utiliser qu'un renseignement, la présence ou l'absence de telle notion dans le texte de départ. Or, le contenu de notre fichier inversé est beaucoup plus riche puisque pour chaque localisation il précise la place du mot dans la phrase, de la phrase dans le paragraphe et de ce dernier dans l'arrêt. Il nous est donc parfaitement possible de faire porter l'interrogation à différents niveaux, les résultats pouvant être très différents. C'est ainsi que l'interrogation au niveau de l'arrêt SECURITE SAUF SOCIALE éliminera tous les arrêts où il est question de Sécurité Sociale. Par contre, la même question au niveau phrase devra être comprise comme éliminant toute phrase contenant les deux mots SECURITE

et SOCIALE. Si un arrêt contient une telle phrase et possède aussi le mot SECURITE non accompagné de SOCIALE dans une autre phrase, cet arrêt sera aussi une solution....

Nous possédons, d'après le contenu du fichier inversé, 4 niveaux possibles d'interrogation : arrêt, paragraphe, phrase, place des mots. Nous utilisons effectivement les trois premiers. Le dernier n'offre aucune difficulté technique pour être mis en oeuvre et il n'est pas exclu que nous le fassions un jour (c'est du reste la raison pour laquelle nous l'avons conservé dans le fichier inversé). Mais la structure des textes d'origine, en phrases très courtes, ne nécessite guère cette possibilité. Comme nous n'en avons guère l'utilisation effective, nous n'avons pas introduit ce niveau de façon à ne pas alourdir le système.

Si l'interrogation peut être ainsi faite à différents niveaux, il est bien évident que l'ensemble d'une question ne va pas être nécessairement au même niveau. C'est ainsi par exemple qu'une interrogation à propos de cotisations de Sécurité Sociale nécessite de chercher SECURITE ET SOCIALE dans la même phrase et COTISATION quelque part dans l'arrêt, mais pas nécessairement dans la même phrase. Nous avons appelé groupe un ensemble de notions liées par des opérateurs logiques au même niveau. Le traitement d'un groupe, c'est-à-dire, la recherche de sa solution, est indépendant du reste de la question. Il constitue donc comme une petite question et sa solution peut être considérée comme une réponse intermédiaire.

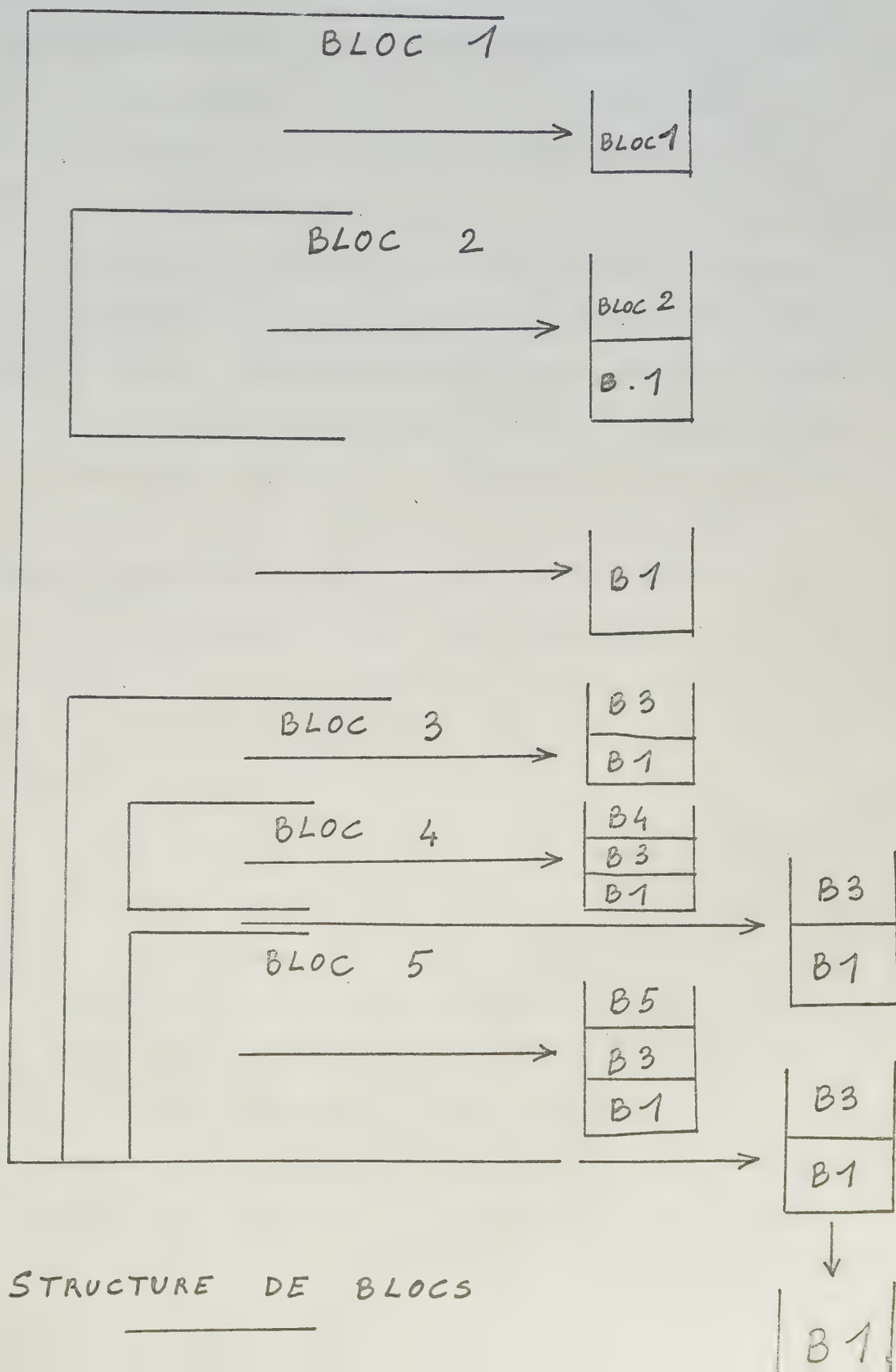


Fig. 24- STRUCTURE DE BLOCS

Les différentes solutions des groupes constituant une question vont être combinées entre elles à l'aide des trois mêmes opérateurs ET, OU et SAUF. Mais elles ne le seront qu'à un seul niveau, l'arrêt. En effet, la solution d'un groupe est constituée par un ensemble d'arrêts et non plus par un ensemble de localisations.

Normalement, les solutions des groupes sont combinées entre elles dans l'ordre d'apparition des groupes. Il peut arriver, dans le cas de questions assez complexes, qu'il y ait intérêt à pouvoir modifier cet ordre de traitement. Pour ce faire, nous avons introduit la notion de bloc. Un bloc, défini par un début de bloc et une fin de bloc, est constitué d'un certain nombre de groupes. Il est traité comme une question entièrement indépendante et a donc une solution elle aussi indépendante du reste de la question. Cette solution peut elle-même être combinée à la solution d'un autre bloc à l'aide d'un des trois opérateurs.

Tout se passe en fait comme si les blocs permettaient de mettre les groupes entre parenthèses, le début de bloc correspondant à la parenthèse ouverte qui serait précédée de l'opérateur à appliquer à ce bloc et la parenthèse fermée serait la fin de bloc. De même qu'il est possible dans l'écriture d'une formule d'avoir plusieurs niveaux de parenthèses, là aussi il est permis d'avoir plusieurs niveaux de blocs. On a des blocs "emboîtés" dont les solutions successives sont placées dans une pile refoulée. L'ouverture d'un bloc (fig. 24) provoque

le stockage sur le dessus de la pile de la solution du bloc courant tandis que la fermeture d'un bloc provoque la combinaison de la solution de ce bloc avec celle située sur le dessus de la pile qui se déplacera d'un cran.

A l'intérieur d'un groupe, l'ordre de traitement est normalement de gauche à droite, sans priorité relative d'un opérateur sur un autre. Dans notre application, nous n'avons, en effet, trouvé aucune justification valable permettant d'attribuer la prépondérance à un des opérateurs. Mais là encore, il peut être utile de pouvoir modifier cet ordre. Supposons qu'on cherche dans le même paragraphe soit PREJUDICE soit DOMMAGE accompagné de soit INDEMNITE soit DEDOMMAGEMENT. Comme il y a quatre combinaisons possibles, il faudrait faire au moins deux groupes liés par OU, chacun étant au niveau paragraphe :

PREJUDICE OU DOMMAGE ET INDEMNITE

et

PREJUDICE OU DOMMAGE ET DEDOMMAGEMENT

Ceci risque d'obliger à en faire un bloc et de toute façon nécessitera deux accès aux localisations des deux premières notions.

Pour répondre à ce besoin, nous avons introduit la notion de sous-groupe, matérialisé par des parenthèses. Cette notion assez restrictive (un seul niveau de parenthèses et le seul opérateur OU à l'intérieur) dans les trois premières versions du programme de recherche a été généralisée sans restrictions dans la 4ème version, pour tous les groupes à un niveau

inférieur à l'arrêt.

En résumé, une question va être composée d'un ou plusieurs blocs, librement emboîtés, chaque bloc étant composé d'un certain nombre de groupes chacun d'eux travaillant à un niveau déterminé et pouvant à son tour être subdivisés en sous-groupes, éventuellement "emboîtés".

-o-o-o-o-o-o-

Chapitre X

LES VERSIONS 1 ET 2 DU PROGRAMME DE RECHERCHE

Le programme de recherche que nous avons appelé D7 est sans doute la pièce la plus complexe et la plus intéressante de tout le système. Nous en possédons actuellement 5 versions successives offrant de plus en plus de possibilités, mais aussi de complexité. Nous allons exposer ici le contenu des deux premières versions.

La version 1 présentait des possibilités assez réduites. Elle a plutôt servi de banc d'essai pour la mise au point de la méthode d'accès au fichier de recherche et pour l'apprentissage des techniques d'interrogations. Ce programme acceptait des questions introduites sur cartes, à raison d'un groupe par carte. Comme nous le verrons plus loin, les éléments constitutifs des groupes devaient être donnés exclusivement à l'aide des numéros de notion. Pour pouvoir interroger, il était donc nécessaire de consulter le lexique.

Les groupes pouvaient être à chacun des trois niveaux (arrêt, paragraphe ou phrase). Ces groupes pouvaient être munis de sous-groupes mais à un seul étage et avec le seul opérateur OU à l'intérieur. Cette version ne connaissait pas la notion de bloc. Enfin, la sortie des résultats ne s'effectuait

QUESTION E3 BIBENT

REPONSE FINALE ** REPONSE QUESTION E3

6540323110290465*STE D AUTOBUS BAYONNE BIARRITZ/SAINT AUBIN*CONTRAT DE TRAVAIL/C
65403231200NGEDIEMENT/RUPTURE ABUSIVE/FAUTE DE L EMPLOYEUR/NECESSITE/LEGERETE BL
6540323130AMABLE/IMPUTATION D UNE FAUTE GRAVE SANS VERIFICATION*

6540786110171165*SA DES ETS BARBIER BENARD ET TURENNE/SPREMONT*CONTRAT DE TRAVAI
6540786120L/CONGEDIEMENT/INDEMNITES/DELAI CONGE/CONDITIONS/TRAVAIL DJ SALARIE/PE
6540786130NDANT LE DELAI CONGE/MODIFICATION UNILATERALE DU CONTRAT PAR L EMPLOYE
6540786140UR/AFFECTATION A UN EMPLOI DIFFERENT*
6540786210CONTRAT DE TRAVAIL/CONGEDIEMENT/RUPTURE ABUSIVE/FAUTE DE L EMPLOYEUR/N
6540786220NECESSITE/INEXACTITUDE DU MOTIF DU RENVOI/ABSENCE D INTENTION MALVEILLA
6540786230NTE DU DE LEGERETE BLAMABLE*

6640324110280366*SA JAMPER/PERNA*CONTRAT DE TRAVAIL/CADRES/APPARTENANCE AUX CADR
6640324120ES/PREUVE/FONCTIONS DE DIRECTION/METALLURGIE*
6640324210 CONTRAT DE TRAVAIL/CONGEDIEMENT/RUPTURE ABUSIVE/FAUTE DE L EMPLOYEUR/
6640324220NECESSITE/MALADIE DU SALARIE/ACCIDENT DU TRAVAIL/LICENCIEMENT APRES LA
6640324230 REPRISE DU TRAVAIL/LEGERETE BLAMABLE*

6640367110220466*ETABLISSEMENT BARBET/ROQUES*CONTRAT DE TRAVAIL/CONGEDIEMENT/RUP
6640367120TURE ABUSIVE/FAUTE DE L EMPLOYEUR/NECESSITE/LEGERETE BLAMABLE/INCONSTA
6640367130NCE DE L EMPLOYEUR/LICENCIEMENT DU SALARIE SANS MOTIF FONDE APRES UNE
6640367140LONGUE PERIODE D ESSAI*
6640367210 CONTRAT DE TRAVAIL/CONGEDIEMENT/DELAI-CONGE/DUREE/USAGES LOCAUX/CONST
6640367220ATIONS NECESSAIRES*

Fig. 25. Réponse à une question avec édition des titres principaux par le programme D7V2.

que sous forme des références des arrêts composant la solution. Ce programme n'utilisait donc en entrée que deux fichiers sur disque, le fichier inversé et le fichier (à un seul enregistrement) de la table de correspondance entre les références internes (binaires et séquentielles) et externes (année, chambre, N°) des arrêts.

Cette version 1 fut en quelque sorte transitoire et ne fut guère utilisée que six mois. Nous sommes passés alors à la version 2, déjà plus élaborée et qui servit à toutes les premières études effectuées et publiées par les juristes de l'équipe avant 1970.

Cette version reprenait tous les dispositifs de la précédente mais en y ajoutant :

- Les blocs , nécessitant l'utilisation d'un fichier de travail sur disque, d'un volume de quelques cylindres.

- L'utilisation du fichier des textes, permettant l'édition des titres principaux en réponse à une question. Cette édition était très élémentaire dans sa présentation, consistant simplement en images de cartes (fig. 25). Nous avons eu un peu tendance à considérer cette possibilité comme un "gadget" .

Les questions sont posées sur cartes. On utilise certaines abréviations :

- 1) - Pour les niveaux d'interrogation, arrêt est symbolisé par A, paragraphe par P et phrase par F.

A 11580E27820
EP 1637 E15381S32830
OF*25360E(2178 038250016815)E(13052017520
SA 25450E21912042510

EB

OB

SB

E

C COMMENTAIRE OU TITRE

* REPONSE PROVISOIRE

** REPONSE FINALE PAR NUMEROS

*** REPONSE FINALE PAR TEXTES

Fig. 26. Exemple de cartes
pouvant composer une question pour DZV2.

2) - Pour les trois opérateurs, on utilise E pour ET, O pour OU et S pour SAUF.

Dans une question, on va trouver plusieurs types de cartes que nous allons étudier un à un (fig. 26).

- Carte de groupe : C'est en quelque sorte la carte principale. Elle comprend 4 parties :

1) - En colonne 1, on doit trouver l'opérateur à appliquer à ce groupe. Si ce groupe est le premier d'une question ou d'un bloc, cette colonne doit rester vierge.

2) - En colonne 2, on place l'indication du niveau d'interrogation pour ce groupe.

3) - En colonne 3, on peut placer un astérisque. Si c'est le cas, la solution du groupe ayant été trouvée, elle sera éditée sur l'imprimante, exclusivement sous forme de références et indépendamment de la solution du bloc ou de la question. Ceci permet, pour une question n'ayant pas donné le résultat escompté, de voir quelle a été l'influence de chacun des groupes sur le résultat final de la question.

4) - A partir de la colonne 4, on trouve la description du groupe lui-même. Chaque notion doit être exprimée sous forme d'un nombre de 5 chiffres (même s'il s'agit d'une notion inférieure à 10000 provenant d'une forme numérique pure qui doit être alors complétée à gauche par des zéros. Les sous-groupes sont matérialisés par des parenthèses.

Il existe enfin une possibilité supplémentaire que nous avons seulement évoquée dans le chapitre IV (p. 30). C'est l'utilisation des notions générales. Nous avons alors signalé que les 4 premiers chiffres du N° de notion étaient caractéristiques de la notion générale. Dans un groupe, si on désire interroger sur une notion générale et non sur une notion spécifique, il suffit de ne donner que les 4 premiers chiffres du N° de notion suivis d'un espace. Une telle notation est équivalente à l'écriture d'un sous-groupe comportant toutes les notions spécifiques de cette notion générale. Par exemple :

EA 1333 E2543Ø est équivalent à (le chiffre 0 étant barré):

EA (1333Ø0133310133320133330.... 013339)E25443Ø

mais avec une écriture tout de même plus légère. Ceci signifie que les localisations de toutes ces notions spécifiques (13330 à 13339) seront fusionnées avant d'être combinées avec celles de 25430.

- Carte d'ouverture de bloc : Cette carte n'utilise que deux colonnes et peut contenir n'importe quoi dans les suivantes.

1) - Colonne 1, l'opérateur à appliquer au bloc.

2) - Colonne 2, un B (pour Bloc ou Begin).

- Carte de fermeture de bloc : Cette carte n'utilise aussi que deux colonnes et peut contenir ce qu'on veut dans les

autres.

1) - Colonne 1, doit être vierge

2) - Colonne 2, contient un E (pour End, F pour Fin ou Fermeture ne pouvant pas être utilisé puisque déjà pris pour indiquer le niveau phrase ... E ne peut signifier ET que s'il est dans la première colonne ou entre 2 notions).

- Carte de fin de question : Cette carte utilise 3 colonnes et peut comporter un message quelconque dans les colonnes suivantes. Son contenu est listé sur l'imprimante en tête de la réponse à la question. Son contenu est le suivant :

1) - Colonnes 1 et 2, deux astérisques. C'est ce qui caractérise cette carte comme étant une fin de question.

2) - Colonne 3, peut contenir un troisième astérisque. Si c'est le cas, la solution de la question sera éditée à l'aide des titres principaux des arrêts. Dans le cas contraire, seules les références des arrêts seront listées sur l'imprimante.

Cette carte de fin de question a deux effets supplémentaires :

1) - Si lors de son apparition un ou plusieurs blocs n'ont pas encore été fermés, elle provoque leur fermeture en cascade. Ceci permet d'éviter éventuellement un certain nombre de cartes E (et surtout d'en mettre une de trop ou de moins).

2) - Les solutions de la question étant éditées, elle provoque le saut en haut de page sur l'imprimante pour pré-

C QUESTION 1 CATALA

F 16510E16810E24730

JP 12430E15390E24500

* RESULTAT AVANT PRECISION

EP*12770E12790E16590E22670

EP*12880E17710E20510E20771

C UN VA OUVRIR UN BLOC

UB

C UN A OUVERT UN BLOC

F 13920E15310

EP 1037 E1243 E1531

* PREMIERE SORTIE PROVISOIRE

EA 13650016000

SA 18700

E

*** RESULTAT D'UNE QUESTION DE DEUX BLOCS LIES PAR 'OU'

C QUESTION 2 ESSAI

P 1001 E1341 E1679

EF 13920E15310

EF 15950E17370

* SORTIE AVANT SAUF

SA 17360018460

** REPONSE QUESTION 2

C QUESTION 3 ESSAI

F 1013 E1243 E1808

EA 12790E16590

EF 12720E17970

** REPONSE QUESTION 3

Figure 27. Exemples de questions. Noter en particulier la première, composée de deux blocs.

parer la question suivante :

- Carte de sortie provisoire : Elle utilise également trois colonnes et est de même listée sur l'imprimante.

1) - Colonne 1, doit contenir un astérisque.

2) - Colonnes 2 et 3, doivent être vierges.

Cette carte provoque la sortie sur l'imprimante de la solution du bloc en cours, exclusivement sous forme de références et ceci sans aucune influence sur le déroulement du reste de la question. Cette possibilité est particulièrement utile lorsque la question posée risque d'être trop précise pour les textes d'origine. La bonne façon de procéder consiste alors à commencer la question d'une façon générale, d'effectuer une sortie provisoire et enfin d'ajouter les précisions. En cas de silence complet, on pourra en général récupérer tout de même la solution dans cette sortie provisoire.

- Carte commentaire : C'est une carte qui est caractérisée par un C en colonne 1, les colonnes 2 à 80 pouvant contenir n'importe quoi. Le seul rôle d'une telle carte est d'être listée sur l'imprimante au moment même où elle est lue. Elle peut se trouver n'importe où et en n'importe quelle quantité dans une question. Son utilité est essentiellement de permettre l'impression d'un titre au début de chaque question.

Sur la figure 27, on trouvera un exemple de question complexe. C'est une question de démonstration, permettant de lier par un OU deux blocs portant sur des sujets aussi différents

REF: III IAI 68JUN 7
EP# 12770E 1279E 16590E 22670

[illegible]

3FSULIAT GROUP E
EP#12880E17710E20510E20771

63 2	325	63 2	380
UN VA QUVER UN BLUC			
QN A QUVERT UN BLUC			
63 2	727	63 2	627

REPONSE PARTIELLE * PREMIERE SORTIE PROVISOIRE

Year	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025	2026	2027	2028	2029	2030	2031	2032	2033	2034	2035	2036	2037	2038	2039	2040	2041	2042	2043	2044	2045	2046	2047	2048	2049	2050	2051	2052	2053	2054	2055	2056	2057	2058	2059	2060	2061	2062	2063	2064	2065	2066	2067	2068	2069	2070	2071	2072	2073	2074	2075	2076	2077	2078	2079	2080	2081	2082	2083	2084	2085	2086	2087	2088	2089	2090	2091	2092	2093	2094	2095	2096	2097	2098	2099	2100
1990	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025	2026	2027	2028	2029	2030	2031	2032	2033	2034	2035	2036	2037	2038	2039	2040	2041	2042	2043	2044	2045	2046	2047	2048	2049	2050	2051	2052	2053	2054	2055	2056	2057	2058	2059	2060	2061	2062	2063	2064	2065	2066	2067	2068	2069	2070	2071	2072	2073	2074	2075	2076	2077	2078	2079	2080	2081	2082	2083	2084	2085	2086	2087	2088	2089	2090	2091	2092	2093	2094	2095	2096	2097	2098	2099	2100

REPONSE FINALE

63200491160163*PERSON/JAME PERSON*ENQUETE/DECISION ORDONNANT DU REFUSANT ENQUE
632004911201E/APPEL/ARTICLE 258 NOUVEAU DU CODE DE PROCEDURE CIVILE/APPLICATION/A
6320049130APPLICATION DANS LE TEMPS*
63200492101FORCE SEPARATION DE CORPS/CAUSES/ADULTERE/PREUVE/CONSTAT/CONSTAT DE
6320049220POLICE/FORCE PROBANTE/APPELATION SOUVERAINE*
63200493101FORCE SEPARATION DE CORPS/CAUSES/ADULTERE/DOUBLE CONDITION DE L ARTI
63200493201E 232 DU CODE CIVIL/NECESSITE NUN *

6320325110260463*VVE JUUISSEAU/ZELIULI*RESPONSABILITE CIVILE/CHOS ES INANIMÉES/
6320325120AKTICLE 1384 DU CODE CIVIL/RESPONSABILITE DE PLEIN DROIT/EXONERATION/F
6320325130AUTE EXCLUSIVE DE LA VICTIME/CIRCULATION ROUTIERE/PIEDON/TRAVERSEE DE
6320325140LA CHAUSSEE/TRAVERSEE HORS DU PASSAGE CLOUTE*

6320727110131163*GUIDRAWEI78UY#RESPONSABILITE CIVILE/CHOSES INANIMEES/ARTICLE 13
632072712084 DU CODE CIVIL/RESPONSABILITE DE PLEIN DROIT/EXUNERATION/FAUTE EXCLU
632072713CSIVE DE LA VICTIME/CIRCULATION ROUTIERE/PIETON/TRAVERSEE DE LA CHAUSSEE
6320727140E/IKAVEESEE HOKS D UN PASSAGE CLOUTE*

Figure 28 - Vue partielle de la réponse à la question 1 de la figure précédente.

qu'un divorce prononcé à la suite d'un adultère avec preuves et une affaire d'accident de la circulation hors d'un passage clouté... On y trouvera aussi une sortie provisoire au milieu de chacun des blocs et la sortie de la solution d'un des groupes.

Sur la figure 28, on trouvera le résultat de cette même question. Ce n'est en fait qu'une partie de la réponse, mais qui permet d'en avoir une idée. On remarquera en particulier comment les références des sorties provisoires des deux blocs n'ont aucun point commun alors que les références des titres principaux listés comme solution définitive se retrouvent soit dans une de ces sorties soit dans l'autre.

L'utilisation de ce programme, baptisé D7V2, permet aux juristes de se familiariser avec le maniement des trois opérateurs logiques et de tirer les premières règles sur la structure et le contenu que devraient avoir les textes d'origine pour pouvoir être pleinement exploitables par de telles méthodes.

Chapitre XI

LA REALISATION DU PROGRAMME D7V2

La première précision qu'il nous faut fournir concerne le langage de programmation utilisé. Quels sont les langages possibles sous D.O.S. ?

1) - GAP (ou RPG) : Nous le citons ici pour être complet, mais il est bien évident qu'il ne viendrait pas l'idée de personne de s'en servir pour ce genre de travail...

2) - FORTRAN : Très bien adapté au calcul scientifique, il est totalement déficient en ce qui concerne le traitement non-numérique et donc en particulier des chaînes de caractères. De plus, ses possibilités d'accès à des fichiers sur disque sont très réduites. Il ne pouvait donc pas convenir.

3) - COBOL : Moins mal adapté que le précédent, il présente toutefois l'inconvénient d'être particulièrement lourd pour la manipulation des chaînes de caractères de longueur variable. De plus, il nécessite une occupation de mémoire assez grande.

4) - PL/1 : C'est probablement le plus adapté des langages évolués mis à notre disposition. Malheureusement, lui aussi est très gourmand en place mémoire et pas très performant. De plus, et ceci est beaucoup plus grave, la version de PL/1 livrée sous DOS n'est qu'un sous-ensemble du langage complet, très res-

treint par rapport à ce dernier. Enfin, nous manquions d'expérience dans ce langage...

5) - ASSEMBLEUR DE BASE : Ce n'est plus un langage évolué mais le langage symbolique du 360. Très proche du langage machine (pratiquement chaque instruction machine du programme généré correspondra à une instruction du langage source) il permet de faire effectuer à l'ordinateur absolument tout ce pourquoi il a été prévu. Enfin, et ceci est important, c'est le seul langage qui permette d'effectuer des entrées-sorties par accès physique.

C'est ce dernier langage que nous avons utilisé pour l'ensemble des programmes de notre système. S'il oblige à écrire des programmes origine plus volumineux que leur équivalent en un langage évolué, le programme généré est souvent bien plus concis et par suite nécessite moins de place en mémoire centrale. Les performances à l'exécution peuvent également être plus élevées, le nombre d'instructions à exécuter pouvant être réduit au strict minimum.

Comme nous le verrons dans un instant, il était très intéressant pour les performances de pouvoir se réserver en mémoire centrale 4 zones de travail de 10000 octets chacune. Comme le superviseur du DOS occupe de 10 à 12 K de mémoire, ceci nous imposait de faire tenir le programme lui-même dans 12 K de mémoire environ. Pour y parvenir, il y avait intérêt à ne rien gaspiller. C'est en particulier sur les entrées-sorties que

nous avons fait porter notre effect :

- Pour les disques, nous avons déjà deux fichiers (celui des textes et le fichier inversé) à accès non standard qui nécessitaient donc l'écriture de toutes les instructions nécessaires à leur gestion. Les deux autres fichiers (celui de la table de correspondance des références et celui nécessaire à la pile utilisée par les blocs) auraient pu être atteints à travers les méthodes d'accès logique. Mais dans ces conditions, il aurait fallu faire appel aux sections de contrôle particulières à ces méthodes. Ces dernières étant relativement volumineuses, nous avons décidé de nous en passer également pour ces deux fichiers d'autant plus facilement que les programmes de canal écrits pour la gestion des deux premiers fichiers pouvaient être également utilisés pour les deux derniers.

- Pour toutes les autres unités périphériques (lecteur de cartes et imprimante) nous avons également fait appel à l'accès physique qui pour ce genre d'unité est plutôt plus simple que l'accès logique standard...

Nous pouvons donc dire qu'une première caractéristique du programme de recherche est d'être écrit entièrement en assembleur et en accès physique. Mais il présente une autre particularité, celle d'utiliser la technique dite de "programmation modulaire". Elle consiste à découper le programme en "unités logiques". Chacune d'elles effectue une fonction bien précise,

MDEB	Doit être utilisée une seule fois en tête de programme. Le label doit comporter 5 caractères:les 4 premiers seront concaténés au nom de toutes les sections;c'est le nom du programme.Le 5ème,s'il est égal à la lettre A indi- que que CC doit être assemblée comme CSECT (sinon DSECT).
SDEB	Début d'une section de contrôle.Point d'entrée normal.
SENT	Point d'entrée supplémentaire dans une section.
SFIN	Fin d'une section.C'est en même temps le point de sortie d'où s'effectue le retour vers la section appelante.Si on est dans la section principale DA,cette macro-instruc- tion est équivalente à EOJ (retour vers le superviseur).
SAPP	Appel d'une section depuis une autre.Le paramètre de cette macro-instruction peut être soit le label d'une macro SDEB soit celui d'une macro SENT.
SOUV	Cette macro-instruction est analogue à la macro standard OPEN,mais permet d'accéder à des fichiers définis dans une autre section de contrôle.
SFER	Analogue à CLOSE,pour des fichiers définis dans une autre section de contrôle.
SLIR	Analogue à la macro standard GET.Le fichier,ainsi que la zone de travail éventuelle,doivent être définis dans une autre section.
SECR	Même chose,mais analogue à PUT.
EXCPV	Identique à la macro standard EXCP,mais portant sur un CCB défini dans une autre section.
WAITV	Identique à WAIT avec la même remarque.

Figure 29.LES MACROS-INSTRUCTIONS PARTICULIERES UTILISEES
POUR LE DECOUPAGE DES PROGRAMMES EN SECTIONS DE CONTROLE

sans interférences avec celle des autres unités. De plus, elles sont traitées comme des sections de contrôle et de ce fait, sont même indépendantes du point de vue de l'assemblage.

Une telle réalisation des programmes, surtout s'ils sont assez complexes, rend deux sortes de service :

1) - Lors de la mise au point, on peut tester séparément chaque morceau. Lorsque l'un d'entre eux l'a été complètement, il n'y a plus à y revenir, quelles que soient les erreurs qui peuvent encore subsister dans d'autres. De plus, les réassemblages souvent nécessaires en cours de tests sont limités à celui d'une seule section de contrôle, d'où gain de temps.

2) - Lors des modifications, si le découpage logique du programme a été judicieusement fait, une modification, même importante, a des chances de ne porter que sur une ou deux des sections. On pourra même dans certains cas remplacer complètement une section par une autre de traitement totalement différent mais acceptant les mêmes entrées et fournissant les mêmes sorties sans qu'il soit nécessaire de toucher au reste du programme.

Il est facile de se rendre compte que dans notre cas, une telle technique était d'un intérêt primordial puisque nous avons réalisé 5 versions successives du même programme. Pour faire le "sectionnement" du programme, nous avons utilisé des macros-instructions particulières mises au point par un collègue qui nous les avait communiquées. On trouvera figure 29, la liste et

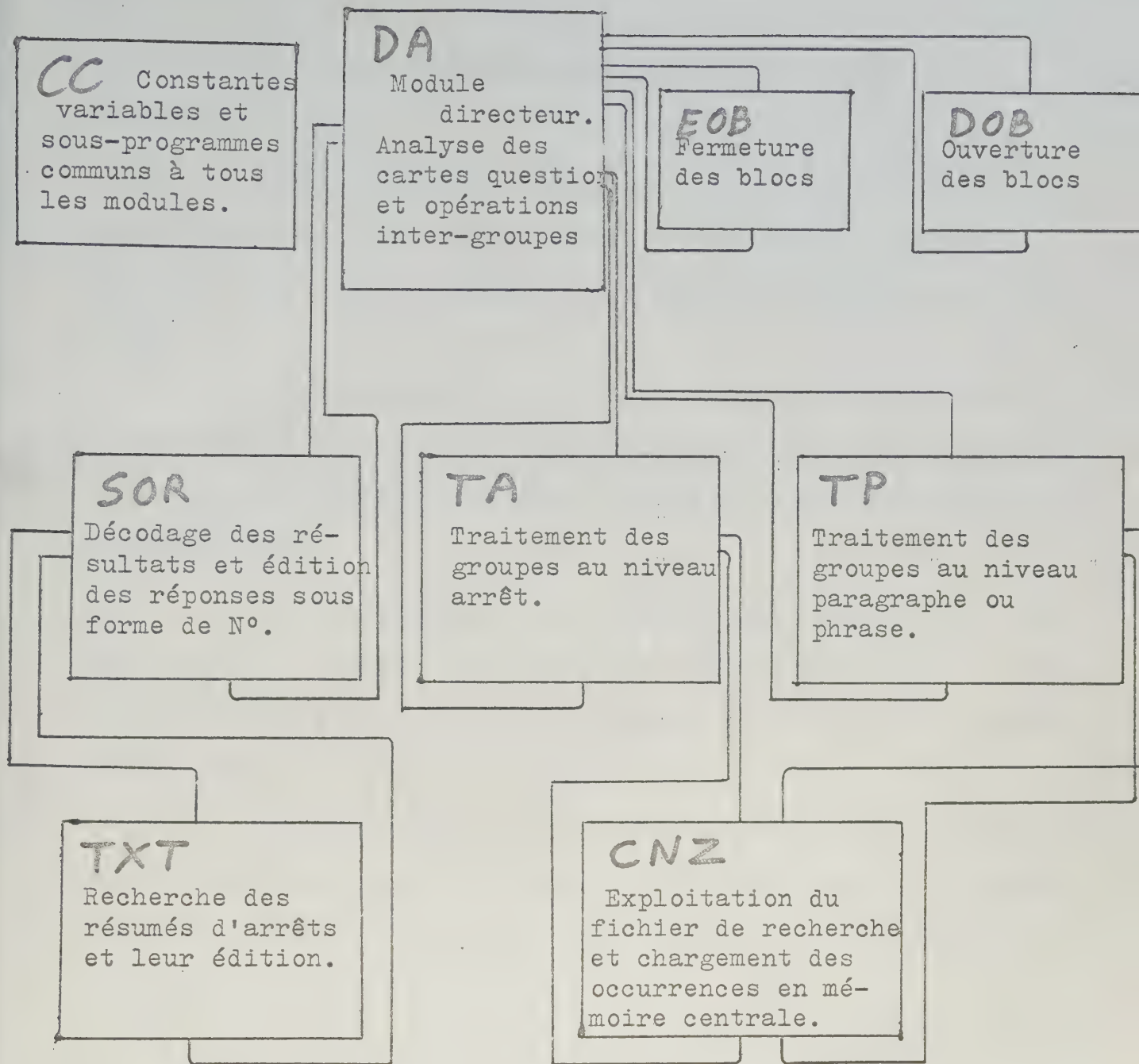


Fig. 30- SCHEMA GENERAL DE L'ORGANISATION DES MODULES
DANS LE PROGRAMME DE RECHERCHE

Chaque module forme une CSECT indépendante et a été assemblé à part.

Le traitement des fichiers disques est effectué par:

- DA pour la table des N° d'arrêts.
- DOB en sortie et EOB en entrée pour les zones de stockage
- CNZ pour le fichier de recherche.
- TXT pour les résumés d'arrêts.

la fonction de ces macros-instructions. Elles utilisent les conventions suivantes :

- Tout programme comprend un "module directeur" qui a le contrôle au début du programme, initialise les registres de base, etc... Il est formé par la section obligatoirement appelée DA.

- On peut avoir une section des constantes et variables communes appelée CC, accessible depuis n'importe quelle autre section. Elle est adressée à l'aide du registre 12 comme registre de base.

- Les différentes sections peuvent s'appeler les unes les autres en cascade. Tous les registres utilisés dans une section sont "sauvegardés" lors de l'appel et restaurés à leur valeur lors du retour.

- Le registre 13 sert à l'adressage de la zone de sauvegarde des registres, le registre 11 sert de registre de base à la section qui a le contrôle, les registres 14 et 15 servent au passage proprement dit.

Le découpage du programme est le suivant (fig. 30) :

- CC : la section des constantes communes comprend en outre les CCB des fichiers sur disque ainsi que les programmes canal . Nous avons pu réduire à 4 le nombre des CCB. En effet, nos quatre fichiers disque sont placés sur deux lecteurs seulement, un pour le fichier des textes et un pour les trois autres fichiers. Ces trois derniers peuvent donc, étant sur la même

unité physique, être déclarés sur la même unité logique et par suite être atteint par le même CCB. Il y a tout de même deux CCB par unité logique, suivant que le programme canal à utiliser correspond à une recherche sur identification ou à une recherche sur clef. De même on ne trouve que deux programmes de canal, chacun d'eux correspondant à un des deux types de recherche, le dernier CCW de chaque programme pouvant être remplacé par un CCW spécifique pour l'opération d'entrée-sortie.

On trouve enfin dans cette section, en plus des constantes, un petit sous-programme nommé PDB ("Pose de Bits") utilisé par trois sections (TA, TP et ISFI). Ce sous-programme permet d'obtenir une représentation extrêmement condensée d'une solution. En effet, la solution d'une question, d'un bloc et même d'un groupe est constituée d'un ensemble d'arrêts. Or, chaque arrêt possède une référence interne binaire et séquentielle. Une solution est caractérisée par la présence ou l'absence de chacun des arrêts, ce qui pour chacun d'entre eux peut être codé au moyen d'un seul bit (0 = absence, 1 = présence). Si on désire représenter la solution dans une certaine zone de mémoire, il n'y a qu'à établir une correspondance bi-univoque entre les bits et les arrêts, ce qui est facilité du fait de la nature de notre référence interne. Nous pouvons donc représenter 8 arrêts par octet. Nous avons réservé une zone 10000 octets, ce qui nous permet de "stocker" 80000 arrêts.

Nous tenons à remercier ici Monsieur DE DAIN, programmeur à IBM FRANCE, travaillant sur les systèmes SAGESSE et IRMS

qui nous avait communiqué cette idée utilisée dans le système SAGESSE. Une telle représentation des solutions rend également très aisée la combinaison des solutions entre elles. Si la solution de deux groupes est stockée dans deux zones, on peut combiner directement ces deux zones à l'aide d'instructions machine pour avoir une image de la solution résultante. Suivant l'opérateur logique :

- Pour ET on utilisera l'instruction NC (ET logique) qui permet de traiter en une seule instruction des zones de longueur maximum de 256 octets bit par bit.

- Pour OU on a OC (OU logique).

- Pour SAUF, on n'a pas une instruction machine permettant de la faire directement. Mais on possède l'instruction XC (OU EXCLUSIF ou encore addition modulo 2). Si on combine la solution du 2ème groupe par XC avec une zone entièrement pleine de 1 binaire, on obtient comme résultat l'ensemble complémentaire de la solution. Il ne reste plus qu'à combiner cette solution avec celle du premier groupe avec l'instruction NC (ET).

- DA : est le "module directeur" du programme. Son rôle est relativement réduit. Après avoir ouvert les fichiers sur disque et lu la table de correspondance des références, il aura deux missions :

- 1) - Lire les cartes qui composent les questions, vérifier leur validité, traiter directement les cartes commentaires et appeler les autres sections en fonction du con-

tenu des cartes.

2) - Effectuer les opérations entre les solutions des groupes ou des blocs et stocker le résultat ainsi obtenu. Nous utilisons au total 4 zones de 10000 octets. Le résultat du bloc en cours est stocké dans la 4ème (qui pourrait être remplacée par une zone sur disque au prix d'une diminution des performances) et le résultat du groupe est pris dans la 2ème zone.

- TA : sert au traitement des groupes dont le niveau d'interrogation est l'arrêt. La solution du groupe en cours d'élaboration est placée sous forme de bits dans la zone N° 2, la solution correspondant à chaque nouvelle notion étant stockée sous la même forme dans la zone N°1, les localisations de cette notion étant provisoirement placées dans la zone N° 3.

- TP : est utilisé pour le traitement à un niveau autre que l'arrêt. A ces niveaux plus réduits, il n'est plus possible d'utiliser les trois instructions élémentaires NC, OC et XC ainsi que la représentation des solutions intermédiaires sous forme de bits. On est donc obligé de garder les localisations elles-mêmes et d'effectuer sur elles des comparaisons sur une longueur dépendant du niveau en vue de conserver ou de supprimer telle ou telle localisation. Les localisations de la première notion sont stockées dans la première des zones de 10000 octets. Il est indispensable que toutes ses localisations puissent être contenues dans cette zone. Nous avons là une des principales limitations contenues dans cette version du programme de recherche.

Le nombre maximum de localisations de toute notion, ou de tout sous-groupe, apparaissant dans un groupe travaillant au niveau paragraphe ou phrase est limité à 2000. Nous aurions pu porter ce nombre à 2500 en supprimant le 5ème octet de chaque localisation (place du mot dans la phrase) mais ceci n'aurait constitué qu'une demi-mesure. Cette limitation est supprimée à partir de la version 4 (voir chapitres XLV et XV).

Les localisations de chaque nouvelle notion ou sous-groupe sont placées dans la zone N° 3. Comme à l'extérieur d'un sous-groupe il n'est pas concevable d'utiliser l'opérateur OU à un niveau inférieur à l'arrêt (que signifie en effet de demander telle notion OU telle notion dans la même phrase ? ...) la solution du groupe est nécessairement déjà contenue dans celle de la première notion. Les localisations de cette solution font donc partie de celles déjà stockées dans la première zone et l'application des différentes opérations de la suite du groupe ne pourra avoir pour effet que d'en supprimer. Cette suppression est effectuée par simple "marquage" à l'aide d'un octet de 1 à la place du N° de paragraphe. A la fin, la solution correspondant aux localisations est placée dans la zone 2.

- ISFI : est utilisé à la fois par TA et par TP. C'est cette section qui permet la gestion du fichier inversé et provoque, étant donné un N° de notion, le chargement de ses localisations dans une des zones de 10000 octets. En fait, cette section fait plus que cela puisqu'elle traite automatiquement les

notions générales et les sous-groupes en provoquant la fusion de leurs localisations. Si ces localisations sont plus nombreuses que 2000 et si cette section a été appelée par TA, elle provoque alors le stockage de ces localisations sous forme de bits.

Du point de vue de la gestion du fichier inversé, cette section possède essentiellement trois sous-programmes spécialisés :

- 1) - ENTFI qui permet de retrouver le premier bloc d'une notion.
- 2) - RCLE qui retrouve le début des localisations dans le bloc.
- 3) - DKSEQ qui permet de lire le bloc suivant.

- DOB est la section permettant l'ouverture d'un bloc. La solution du bloc en cours tient sur 10000 octets. Elle est vidée sur disque en trois enregistrements (un par piste). La liste des adresses sur disque du premier enregistrement de chaque étage de bloc est contenue dans la section CC (TBLK) un pointeur (PTBLK) produisant l'effet de pile.

- EOB est la section complémentaire de la précédente. Elle remplace la solution du bloc courant dans la zone N° 2 et recharge dans la N°4 celle du bloc précédent. La combinaison de ces deux solutions est alors effectuée par DA comme s'il s'agissait de celles de deux groupes. EOB est également appelé par DA à la rencontre d'une carte fin de question s'il reste des blocs encore ouverts.

- SOR est spécialisée dans l'édition des résultats. La solution est supposée se trouver dans la zone N°2 sous forme de bits. Un premier décodage permet de passer de cette représentation à la référence interne, un second décodage utilisant la table de correspondance permettant d'avoir la référence externe. Si la sortie doit être effectuée avec les seules références, c'est elles qui sont alors éditées. Sinon, pour chaque référence reconstituée, SOR appelle la section suivante , TXT.

- TXT utilisant le fichier texte, y lit l'information correspondant à la référence passée par SOR et édite le titre principal de cet arrêt.

-o-o-o-o-o-

Chapitre XII

LA VERSION 3 - LE SYSTEME CONVERSATIONNEL

Notre système, muni du programme de recherche version 2, fit l'objet de plusieurs communications lors de Congrès (9ème Congrès de l'European Guide à Naples en mai 68 , "l'Informatique au service de l'Homme" à Grenoble en mars 69, etc...) ainsi que d'exposés effectués sur place pour différents groupes de spécialistes. Or, à la suite d'échanges d'idées survenus dans ces diverses occasions, nous nous sommes aperçus qu'un des principaux reproches fait à notre méthode était l'obligation de coder les questions, c'est-à-dire, essentiellement de ne pouvoir utiliser que les N° de notion dans la rédaction des questions.

En fait, nous ne pensions pas que ce soit réellement un inconvénient grave. Mais devant l'insistance de nos interlocuteurs, nous avons décidé d'étudier le problème. Il est bien évident que la traduction en code d'une question rédigée en clair ne pose aucun problème technique grave à partir du moment où on possède un lexique structuré de telle sorte que son accès soit rapide. Il suffit de poser comme principe que pour avoir une certaine notion dans une question, il suffit d'utiliser une des formes- α associées à cette question.

Une première solution consistait en un programme indépendant qui aurait fourni les questions traduites soit sur cartes (et le programme de recherche n'avait pas à être modifié), soit sur disque pour y être reprises par le programme de recherche. Dans ce cas la modification aurait été mineure et n'aurait porté que sur la section DA. Mais il nous est alors apparu comme étant beaucoup plus intéressant d'introduire cette traduction dans le programme de recherche lui-même en la complétant par une possibilité de fonctionnement en mode conversationnel...

Ce dernier est caractérisé par le fait qu'il s'établit un véritable dialogue entre l'opérateur et l'ordinateur. Cela suppose que l'ordinateur tienne l'opérateur au courant de l'avancement du déroulement du programme et que ce dernier ait la possibilité d'intervenir sur son cours. En particulier si le programme a pour objet le traitement d'une suite de données (comme c'est ici le cas avec les groupes successifs) il est indispensable que ces données soient introduites par l'opérateur lui-même, en fonction de l'avancement du traitement, le contenu même de ces données devant pouvoir varier. Ceci exclut donc l'utilisation des cartes et impose pratiquement l'usage d'un terminal à clavier. Notre installation étant dépourvue de véritable terminal, c'est la machine à écrire du pupitre qui en tient lieu.

Nous avons donc prévu pour cette version du programme de recherche le schéma suivant :

1) - L'utilisateur entre les questions en clair sur la machine à écrire, groupe par groupe. Il utilise pour cela un langage assez souple qui sera décrit plus loin.

2) - Le programme, utilisant un lexique réorganisé, traduit de suite ce groupe et l'exécute. Si au cours de cette traduction ou au cours de l'exécution du groupe, se présentent des difficultés (telle que la présence d'une forme- α inconnue au lexique) l'ordinateur doit de suite en informer l'opérateur qui doit pouvoir choisir une solution. De toute façon, le groupe ayant été traité, il y a lieu d'indiquer sur le pupitre le nombre d'arrêts sélectionnés par ce groupe et le nombre de réponses du bloc compte tenu de ce dernier.

3) - Ayant ces renseignements, l'utilisateur doit alors pouvoir, soit poursuivre son interrogation par un nouveau groupe (ou la demande du résultat final), soit faire une "marche arrière" en annulant l'effet du dernier groupe.

4) - Les résultats, partiels ou définitifs, sont toujours listés sur l'imprimante, leur sortie sur la machine à écrire ne présentant guère d'intérêt (ce serait différent sur un terminal placé à distance) et augmentant le temp d'exploitation.

5) - Sur option, le programme doit pouvoir perforer sur cartes les questions traduites et cela sous une forme qui

REPOSE FINAL

6510019110060165*LANGLOIS BERTHELOT/L ASSISTANCE PUBLIQUE*JUGEMENTS ET ARRETS/ME
65100191200120015*DESIGNATION DE LA JURIDICTION AYANT STATUE/INDICATION SUPPLEE
65100191300130015*LES AUTRES MENTIONS DE LA DECISION*
65100192100120015*TESTAMENT/LEGS/LEGATAIRE/DESIGNATION/CEUVRE CHARITABLE/CEUVRE NON DE
65100192200120015*RECHERCHE DE L INTENTION DU TESTATEUR*
65100193100130015*INFORMATION/MOTIFS/REFUTATION DES MOTIFS DU JUGEMENT ENTREPRIS/N
65100193200130015*NON *

REPOSE FINALE ***FIN***

65 1 0019 C6/C1/65
LANGLOIS BERTHELOT/L ASSISTANCE PUBLIQUE
JUGEMENTS ET ARRETS/MENTIONS/DESIGNATION DE LA JURIDICTION AYANT STATUEE/INDICATION SUPPLEE PAR LES AUTRES
MENTIONS DE LA DECISION
TESTAMENT/LEGS/LEGATAIRE/DESIGNATION/CEUVRE CHARITABLE/CEUVRE NON DETERMINEE/RECHERCHE DE L INTENTION DU
TESTATEUR
APPEL/INFORMATION/MOTIFS/REFUTATION DES MOTIFS DU JUGEMENT ENTREPRIS/NECESSITE NON

c question 1
BG phrase oeuvre et charitable
BG SOLUTIONS GROUPE 2
BG et phrase intention et testateur
BG SOLUTIONS GROUPE 2
BG SOLUTIONS BLOC 1
BG fin
BG

Figure 30 bis - Résultat de la même question avec la version 2 (en haut) et la version 3 (en bas) -
Ci-contre, formulation de la question au pupitre en version 3 -

soit acceptée en entrée. Il est donc nécessaire que ce programme puisse être utilisé également à partir de cartes. Dans le cas où l'interrogation doit porter successivement sur plusieurs fichiers (c'est le cas sur notre installation à cause du fichier des textes, un chargeur de disque ne permettant le stockage que de 5 années) les questions ayant été mises au point en mode conversationnel sont alors exploitées sans perte de temps sur les fichiers suivants.

Le programme de recherche version 3, D7V3 , répond exactement au schéma ci-dessus. En outre, nous avons profité de l'occasion pour remanier la présentation des titres principaux édités comme solution (fig. 30 bis) l'intérêt d'une telle édition nous étant clairement apparu entre temps. Cette nouvelle présentation, non seulement supprime les dix premières colonnes des images de cartes mais encore fait en sorte que les mots ne soient pas coupés et apparaissent en entier sur une ligne.

Nous allons voir à présent comment doit être introduite une question en mode conversationnel. Lors de l'appel du programme, le mode conversationnel est activé si la première carte présente sur le lecteur est une carte /* (fin de données). La machine ayant le contrôle, on doit introduire un premier message précisant les options :

1) - La présence d'un C dans ce message (qui peut être de longueur quelconque) signifie qu'on désire la perforation simultanée des questions traduites.

2) - La présence d'un W demande que l'analyse d'un groupe soit effectuée en cours de frappe, sa traduction ayant lieu simultanément à son introduction. La vitesse d'exécution est alors un peu augmentée, au détriment d'une augmentation importante du temps "compteur".

Ces deux options sont indépendantes l'une de l'autre. Pour la suite de la question, 5 règles générales sont applicables :

1) - La frappe peut être faite indifféremment en majuscules ou en minuscules.

2) - Les "mots clés" du langage comportant plus de 4 lettres peuvent être tronqués à 4. Les caractères supplémentaires seront ignorés. De plus, certains d'entre eux, en particulier les opérateurs, présentent aussi une forme abrégée à un seul caractère.

3) - Certains de ces mots-clés sont optionnels, des options par défaut étant prises en leur absence.

4) - Les délimiteurs sont constitués par tout signe inférieur à une lettre, excepté le signe - qui sert dans les mots composés. On peut intercaler autant d'espaces qu'on désire entre les termes.

5) - En cas d'erreur de frappe, on peut annuler les derniers caractères frappés à l'aide du signe \$. Plusieurs \$ à la suite annulent le même nombre de caractères précédant immédiatement le premier \$. C'est ainsi que JUGGE\$\$EMENT sera

interprété comme JUGEMENT.

Dans ce qui suit, nous représenterons les mots-clés du langage en majuscules pour les 4 premiers caractères, les suivants étant écrits en minuscules pour signifier qu'ils ne sont pas nécessaires. Ceci n'indique pas comment il faut effectivement les écrire puisqu'ils peuvent l'être aussi bien en majuscules qu'en minuscules.

- Ecriture des groupes

Les trois opérateurs peuvent revêtir les formes suivantes :

- pour ET : E ET

- pour OU : O OU , (cette dernière représentation permettant en particulier d'alléger l'écriture des sous-groupes).

- pour SAUF : S SAUF

Les trois niveaux sont indiqués par :

A ou ARREt , P ou PARAgraphe et F ou PHRAse

En tête du groupe, s'il n'est pas le premier bloc, on peut trouver un opérateur. Si ce groupe n'est pas le premier et si l'opérateur est omis, la valeur prise par défaut sera ET. En effet, il nous est apparu que c'était cet opérateur qui est le plus fréquent.

Après l'opérateur, ou en tête dans le cas du premier groupe d'un bloc, on peut trouver le niveau. Là encore cette indication est optionnelle, la valeur par défaut étant

ARRET.

Les notions peuvent être entrées : soit sous forme numérique en 1 à 5 chiffres, soit à l'aide d'une des formes- α associée à cette notion. Si l'interrogation doit porter sur la notion générale, il suffit de faire suivre la forme- α de /G.

Lors de la traduction d'une forme en son N° de notation, il peut arriver que le programme trouve qu'elle n'existe pas au lexique.

Le message suivant apparaît alors :

PAS DE NOTION POUR XXXXXXXX

L'opérateur a alors le choix entre deux actions possibles :

1) - Frappe d'une nouvelle notion, sous une des deux formes.

2) - Introduction du mot RIEN qui aura pour effet de supprimer cette notion et son opérateur.

Le début d'un sous-groupe est toujours indiqué par (et sa fin par) . Dans ce domaine, les restrictions d'usage sont les mêmes que pour la version 2. A l'intérieur d'un sous-groupe, il y a élimination automatique des notions en double (on peut très bien avoir indiqué deux formes- α distinctes sans savoir qu'elles ont le même N° de notion).

Si on désire l'édition de la solution du groupe, il faut faire suivre ce dernier de *Sortie. La fin de l'introduction du groupe est matérialisée par la touche "fin de bloc".

C QUESTION 15

RÉPONSE FINALE

***FIN**

66 1 0164 07/03/66

COMPAGNIE D ASSURANCES LONDON ET LANCASHIRE COMPANY LIMITED/CONSORTS HOUE ET AUTRES
RESPONSABILITE CIVILE/FAUTE/EXPLOSION/BOUTEILLE/GAZ/MULTIPLICITE DES CAUSES POSSIBLES/IMPUTABILITE AU VE

68 2 0304 11/12/68

DORGANS/LAJASSE

RESPONSABILITE CIVILE/CHOSSES INANIMEES/ARTICLE 1384 DU CCDE CIVIL/GARDIEN/PROPRIETAIRE/BOUTEILLES A GAZ
COMPRIME/RECHARGEMENT/EXPLOSION

RESPONSABILITE CIVILE/CHOSSES INANIMEES/ARTICLE 1384 DU CCDE CIVIL/RESPONSABILITE DE PLEIN DROIT/EXONERATION
CAS FORTUIT OU DE FORCE MAJEURE/VICE INHERENT A LA CHOSE

c question 15

BG arrêt propane

BG PAS DE NOTION POUR PROPANE

butagaz

BG A 12210

BG NOTION NON TROUVEE DANS LE FICHIER 12210
combustible

BG PAS DE NOTION POUR COMBUSTIBLE

gaz

BG SOLUTIONS GROUPE

6

BG et explosion

BG SOLUTIONS GROUPE

6

BG SOLUTIONS BLOC

2

BG fin

BG

Figure 31- Remplacements successifs d'une forme-d par une autre
et d'une notion par une autre.

C BOUTEILLE DE GAZ
NOTION NON TROUVEE DANS LE FICHIER 53160 - IGNOREE.

REPONSE FINALE ***FIN*

65 2 0144 11/02/65

STE VALLOUREC/MACARI

CHOSE JUGEE/AUTORITE DU PENAL/HOMICIDE OU BLESSURES INVOLONTAIRES/RELAXE/PORTEE/ARTICLE 1382 DU CODE CIVIL/
RELAXE DU DIRECTEUR TECHNIQUE D UNE SOCIETE FABRIQUANT DES BOUTEILLES A GAZ/ACTION CIVILE DIRIGEE CONTRE LA
SOCIETE/POSSIBILITE

66 1 0164 07/C3/66

COMPAGNIE D ASSURANCES LONDON ET LANCASHIRE COMPANY LIMITED/CONSORTS HOUE ET AUTRES

RESPONSABILITE CIVILE/FAUTE/EXPLOSION/BOUTEILLE/GAZ/MULTIPICITE DES CAUSES POSSIBLES/IMPUTABILITE AU VENDEUR

66 2 0350 16/C3/66

SOCIETE D APPLICATION DES GAZ CAMPING GAZ/CONSORTS FRANCOIS PIALAT

RESPONSABILITE CIVILE/FAUTE/FABRICANT/BOUTEILLE DE GAZ/VICE DE CONSTRUCTION/INFLAMMATION DU GAZ AU CONTACT
D UNE SOURCE D INCANDESCENCE/ORIGINE DE CELLE CI NON ETABLIE/EFFET

68 2 0304 11/12/68

DORGANS/LAJASSE

RESPONSABILITE CIVILE/CHOSES INANIMEES/ARTICLE 1384 DU CODE CIVIL/GARDIEN/PROPRIETAIRE/BOUTEILLES A GAZ

COMPRIME/RECHARGEMENT/EXPLOSION

RESPONSABILITE CIVILE/CHOSES INANIMEES/ARTICLE 1384 DU CODE CIVIL/RESPONSABILITE DE PLEIN DROIT/EXONERATION/
CAS FORTUIT OU DE FORCE MAJEURE/VICE INHERENT A LA CHOSE

Figure 31 bis - Suppression d'une notion sans localisation.

c bouteille de gaz

BG bouteille et (gaz ou butane)

BG A 12080E(17390053160)

BG NOTION NON TROUVEE DANS LE FICHIER 53160

Rien

BG SOLUTIONS GROUPE 4

BG fin

BG

Si un incident survient en cours de traitement du bloc, un message apparaît (fig. 31 et 31 bis) comportant deux parties :

- la traduction du groupe sous la forme utilisée en version 2.
- un message explicatif.

Là encore, l'opérateur a la possibilité de remplacer la notion en cause (c'est-à-dire celle en cours de traitement) par une autre ou de la supprimer par le mot RIEN.

Le groupe ayant été entièrement traité, le programme retourne deux messages :

- le premier donne le nombre de documents qui sont des solutions du groupe lui-même.
- le suivant l'indique pour le bloc, en cours de traitement.

Si l'opérateur trouve alors que l'action de ce groupe a été néfaste, il a la possibilité de l'annuler et entrant RETOUR. Dans ce cas, la solution du bloc redeviendra ce qu'elle était avant introduction du groupe et de plus si on utilise l'option de perforation simultanée des questions traduites, la carte correspondant à ce groupe ne sera pas perforée.

Si par contre l'opérateur désire conserver l'effet de ce groupe, il lui suffit de commencer son prochain message par autre chose que RETOUR.

- Ouverture d'un bloc. On peut commencer par indiquer l'opérateur à appliquer au bloc. Par défaut, ce sera ET. Le deuxième mot du message doit être ensuite soit OUVerture soit DEBUt.

- Fermeture de bloc. Elle s'obtient soit à l'aide du seul terme FERMeture, soit à l'aide des deux mots FIN BLOC.

- Sortie provisoire. Elle s'indique par SORTie PROVi-soire.

- Fin de question. Le premier terme peut être FIN ou SORTie. S'il est seul ; on obtiendra la sortie sous forme des titres principaux de même si le terme suivant est TEXTes. Par contre on l'aura sous forme de références en le faisant suivre de REFERENCEs.

- Commentaires. Toute ligne commençant par C suivi d'au moins un espace sera considéré comme telle.

- Fin de travail (retour au superviseur). Elle s'obtient par le terme TERMINE.

-O-O-O-O-

Chapitre XIII

REALISATION DE D7V3

Il y avait deux optiques possibles pour réaliser le travail. La première consistait à modifier tout ce qui pouvait avoir un rapport plus ou moins lointain avec le mode conversationnel. Mais on peut se rendre compte que la plupart des sections du programme D7V2 risquaient d'être modifiées. L'autre solution consistait au contraire à ne rien modifier sauf la section DA de façon à l'adapter à la nouvelle façon d'introduire les groupes. Il faut alors s'arranger pour faire la traduction complète de chaque groupe avant de commencer le traitement, cette traduction devant donner comme résultat une image de ce qu'on aurait en entrée par cartes avec la version précédente.

C'est cette dernière solution que nous avons retenue. Mais nous avons préféré ajouter encore une section au programme, section spécialisée dans l'acquisition et la traduction des questions en mode conversationnel. Dans DA, nous avons laissé les instructions nécessaires à la lecture des questions sur cartes, mais en les faisant précéder d'un aiguillage positionné en début de programme qui permet soit d'y passer, soit d'appeler à la place la section spéciale. Cette

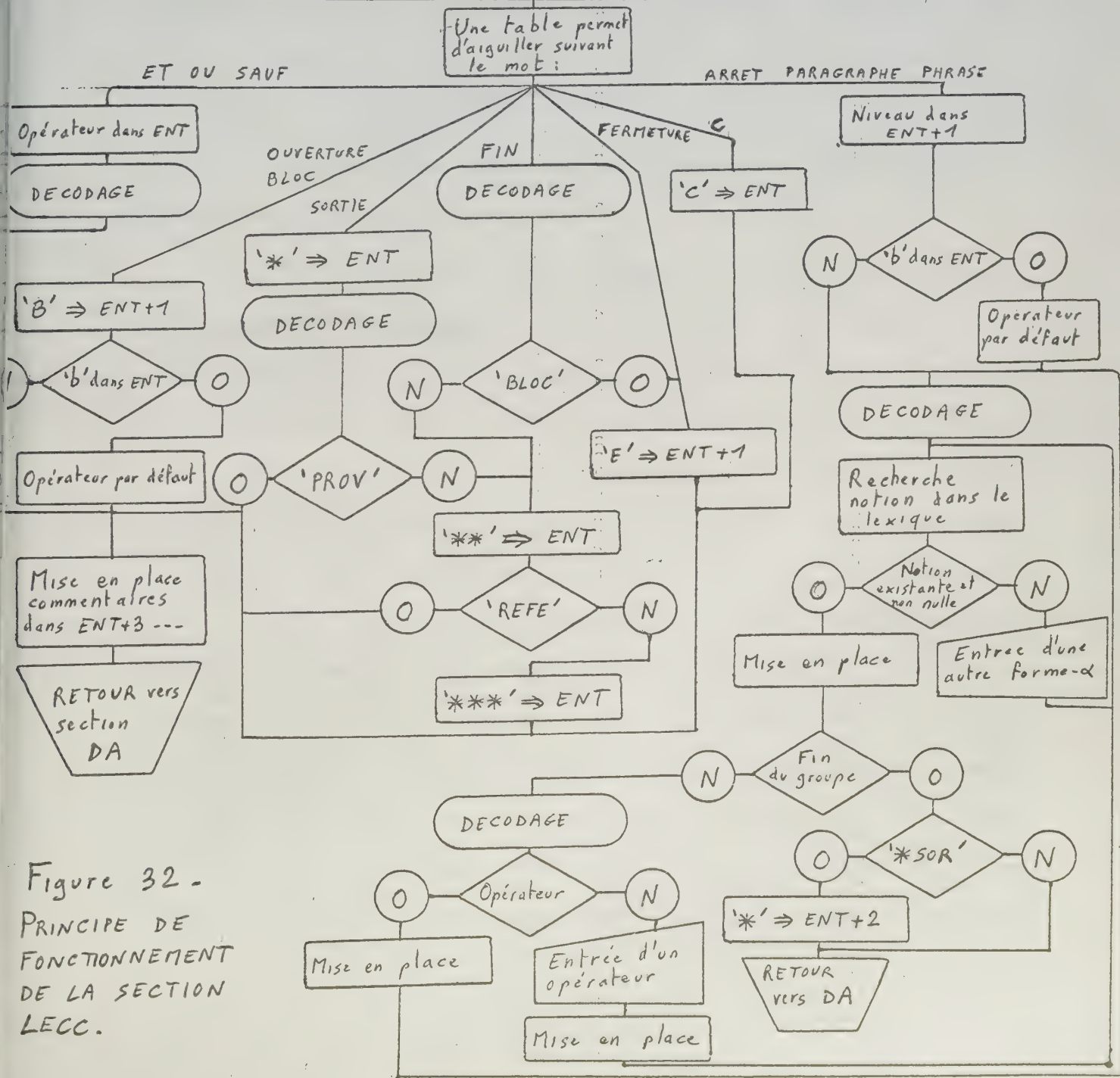
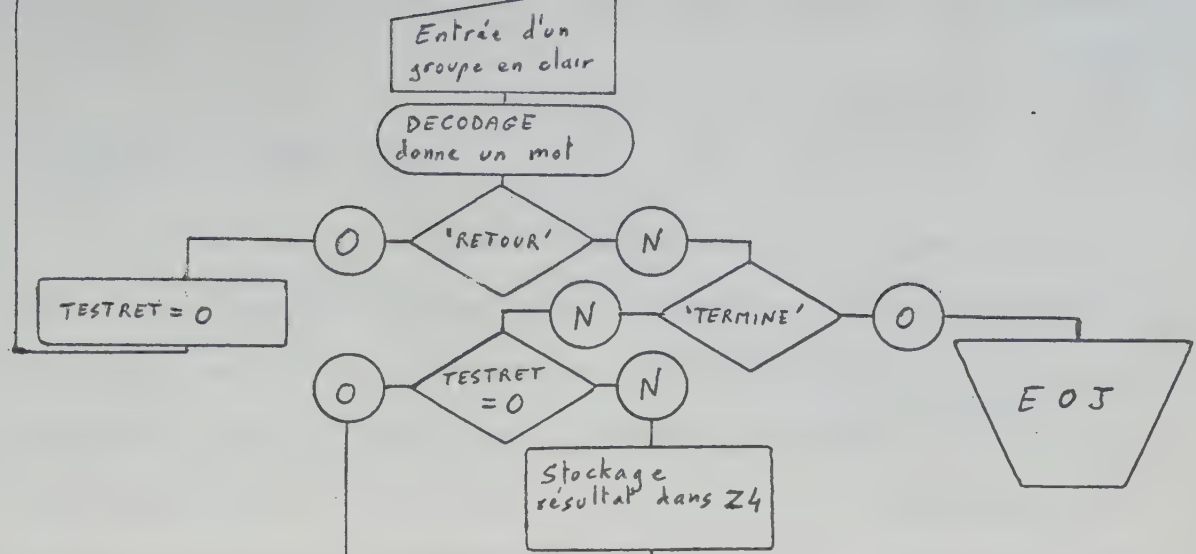


Figure 32 -
PRINCIPE DE
FONCTIONNEMENT
DE LA SECTION
LECC.

dernière se nomme LECC.

En plus de la modification due à l'appel de LECC, il a fallu introduire dans DA le comptage des solutions et l'édition de ce nombre avant et après la combinaison de chaque groupe avec le résultat précédent. De même le stockage de la solution du bloc dans la zone N° 4 n'a lieu dans DA que dans le cas de questions sur cartes et a lieu dans LECC dans le cas contraire. Ceci permet de ne pas l'effectuer si on désire annuler l'effet du dernier groupe.

La section LECC (organigramme fig. 32) présente plusieurs fonctions distinctes.

1) - Lecture sur la machine à écrire d'un message d'une longueur maximum de 1000 caractères. Ce message va être démonté mot par mot. Comme la machine à écrire, surtout en entrée est lente et comme le traitement de certains de ces mots va nécessiter un accès disque, il est possible de jouer sur la simultanéité vraie des entrées-sorties et du calcul. La machine à écrire est sur le canal multiple, les disques sur le canal simple 1, les deux peuvent donc également fonctionner ensemble. Ceci suppose qu'on récupère les lettres du message une à une au fur et à mesure de leur introduction. Pour le réaliser, la zone d'entrée est préalablement remplie de 1 binaire (il est impossible d'introduire par la machine à écrire un octet ayant cette configuration). Un pointeur se déplace sur cette zone et alimente un test comparant un octet

à 8 un binaire . S'il y a égalité , le programme boucle sur le test, sinon le caractère est pris ...

2) - En début de groupe, on doit chercher l'opérateur et le niveau, mais qui peuvent être absents, ou un autre mot-clef au cas où le message ne serait pas un groupe. Il faut donc pouvoir reconnaître les mots-clefs et prendre une action en conséquence. Ceci s'effectue à l'aide d'une table (TABMOT) qui comprend chaque mot-clef suivi de ce qui doit apparaître dans les deux premières colonnes de la traduction du fait de sa présence et de l'indication du sous-programme auquel il faut se rendre si on l'a trouvé.

3) - Si le premier mot ne commence pas par RETO, il faut aussi éventuellement stocker le résultat provisoire dans la zone N°4.

4) - A partir du troisième mot (ou plus tôt si l'opérateur ou le niveau manque) il faut remplacer les formes- α par leur N° de notion. Pour cela, LECC va utiliser un fichier supplémentaire, le lexique réorganisé.

Nous aurions très bien pu utiliser ici la méthode d'accès du Séquentiel Indexé standard. Mais là encore, le désir de gagner de la place nous a fait opter pour une méthode personnelle basée sur le même principe. La clef est donnée par la forme- α elle-même sur 25 caractères. On utilise deux index, un de cylindre et un de piste comme dans le fichier inversé. Mais cette fois, l'index de piste ne comporte qu'un enregistre-

DOS

STMT	SOURCE	STATEMENT	
394		UNFK ZCDEC+10(5),ZCDEC+15(3)	
395		GI ZCDEC+14,X'FO'	
396		LA 9,ZCDEC+10	
397		B CHERCK	
398	PKCCN	PACK ZCDEC+15(3),ZCDEC(1)	
399	ALPHA	CLI ZCDEC,C'	
400		BE MCT10-8	
401		LA 9,3	
402	LD	EXCP DK	COMPTEUR DE BOUCLE
406		WAIT DK	LECTURE
412		L 2,=F'50000'	
413		MVC ID+3(3),0(2)	
414		BCT 9,LC	MODIFICATION IDENTIFICATION
415		MVI TESTAC,0	RETOUR EN LECTURE (2 fois)
416		L 9,=F'50000'	ICI LE BLOC CHERCHE EST
417		LA 2,26	EN MEMOIRE
418	CYCNC	CLC 5(25,9),ZCDEC	
419		BH INCNC	
420		BE CHERCK	
421		LA 9,30(9)	
422		BCT 2,CYCNC	
423		B INCNC	
424	CHERCK	BC 0,RETINC	
425		MVC 0(5,4),0(9)	
426		CLI 0(9),C'0'	
427		BNL NOTG	
428		MVC COT(19),=C' NOT ION	'NUL' POUR
429		B INCNC+6	
430	ID	DS CL7	
431	CYLND	DC X'000000A0000001'	
432	TESTAC	DS C	
433	DK	CCB SYS005,CDK	
444	CDK	CCW 7,ID,X'40',6	SEEK
445		CCW 49,ID+2,X'40',5	RECHERCHE R1
446		CCW 8,*-8,0,1	
447		CCW 105,ZCDEC,X'40',25	RECHERCHE CLE >>
448		CCW 8,*-8,0,1	
449		CCW 6,50000,X'20',780	LECTURE

Figure 33. Lecture du lexique pour la traduction des formes- α en notation.

ment par piste. Chaque bloc de données est de longueur 780 et comporte 26 formes- α ; il possède aussi une clef qui est égale à la dernière forme- α contenue dans ce bloc. La lecture d'un bloc va donc se faire à l'aide de trois accès successifs, tous trois à l'aide d'un SEARCH sur clé égale ou supérieure. Ces trois lectures sont effectuées dans une boucle nommée LD (fig. 33).

Cette section LECC comporte encore plusieurs fonctions moins importantes qu'il serait fastidieux d'exposer ici.

Les autres sections ne nécessitèrent que de très légères retouches, dues en particulier à la nécessité de pouvoir remplacer une notion par une autre en cas d'incident en cours de traitement. Ceci intéressait en particulier les sections ISFI et TP.

Enfin, la section TXT fut entièrement reprise pour permettre une refonte complète de la présentation des titres principaux en sortie. Mais cette modification n'avait aucun lien avec le fait d'introduire le mode conversationnel.

Chapitre XIV

LA VERSION 4 DU PROGRAMME DE RECHERCHE

Tel qu'il vient d'être exposé, c'est-à-dire, au stade de la version 3, le programme de recherche présentait un certain nombre de limitations. Les plus importantes étaient les suivantes :

1) - Un seul niveau de parenthèses à l'intérieur d'un groupe avec le seul opérateur OU à l'intérieur des parenthèses.

2) - Le nombre de localisations d'une notion ou d'un sous-groupe ne pouvait pas excéder 2000 si le traitement du groupe était au niveau phrase ou paragraphe. On doit, en effet, travailler alors directement sur les localisations et dans cette version, ces dernières sont traitées en mémoire centrale. Il était donc nécessaire de pouvoir charger en mémoire les localisations de chaque notion successive dans une zone de 10000 octets.

En fait, ces limitations sont-elles très gênantes ? La première ne le semble pas à première vue et il nous a fallu un certain temps pour nous rendre compte qu'elle l'est. La seconde ne nous inquiétait pas trop au début car nous ne pensions pas avoir à interroger sur des notions trop fréquentes. En fait,

il est bien connu que la valeur informationnelle d'un terme est inversement proportionnelle à sa fréquence. Mais il arrive souvent, tout au moins dans les documents que nous avons traités, qu'il soit nécessaire d'associer à un terme peu fréquent un autre très fréquent dans le seul but de préciser le premier. C'est ainsi que la notion RESPONSABILITE est souvent utilisée en cours d'interrogation alors qu'elle doit être présente dans 25 % des arrêts.

C'est donc dans le but de permettre l'utilisation des notions fréquentes à n'importe quel niveau que nous avons envisagé la mise au point d'une nouvelle version. Il est bien évident que le traitement d'ensembles de données trop volumineux pour être entièrement chargés en mémoire centrale et la production de résultats intermédiaires eux-même trop importants, nécessite l'utilisation de zones de stockage intermédiaire sur disques. Il n'est plus possible de travailler uniquement en unité centrale. Ceci va donc provoquer :

1) - Une augmentation de la complexité du traitement et donc du programme.

2) - Une perte de performance due au temps d'accès aux disques.

C'est donc une modification particulièrement importante qu'il fallait envisager. Mais alors il est apparu comme étant plus judicieux de profiter de l'occasion pour résoudre l'autre limitation, et cela d'autant plus volontiers que la solution

c question 22
 phrase caisse sauf securite sauf assurance
 SOLUTIONS GROUPE 307
 ou phrase caisse et (securite ou assurance) sauf sociale
 SOLUTIONS GROUPE 0
 SOLUTIONS BLOC 307
 et phrase caisse sauf((securite ou assurance) et sociale)
 SOLUTIONS GROUPE 307
 SOLUTIONS BLOC 307
 sortie references

Figure 34. Equivalence d'un groupe à parenthèses
 multiples et de deux groupes ordinaires.

technique est en bonne partie la même pour les deux problèmes (voir le chapitre suivant). Voici donc à présent quel est exactement le problème posé par la première limitation, à savoir la possibilité d'un seul niveau de parenthèses dans un groupe. Il est très différent suivant le niveau d'interrogation.

1) - Au niveau arrêt, c'est un faux problème. En effet, le niveau de combinaison des groupes entre eux étant l'arrêt, on peut toujours considérer plusieurs groupes comme n'en formant qu'un, de même qu'il est toujours possible de couper un groupe en plusieurs autres. C'est ainsi que le groupe :

ARRET ACCIDENT ET CIRCULATION ET PIETON
est très exactement équivalent aux deux groupes :

ARRET ACCIDENT ET CIRCULATION
ET ARRET PIETON

Si on désire placer plusieurs niveaux de parenthèses dans un groupe, il suffit de le décomposer en plusieurs qui seront éventuellement regroupés en plusieurs blocs.

2) - Au niveau paragraphe ou phrase, le problème n'est plus aussi simple. Supposons qu'on désire sélectionner tous les arrêts dans lesquels se trouve une phrase contenant le mot CAISSE non associé à SECURITE SOCIALE ou à ASSURANCES SOCIALES. On peut encore les avoir à l'aide de deux groupes :

PHRASE CAISSE SAUF SECURITE SAUF ASSURANCE
OU PHRASE CAISSE ET (SECURITE OU ASSURANCE) SAUF
SOCIALE.

C QUESTION 9

REPONSE FINALE

***FIN*

68 5 0162 14/C3/68

SS DU JURA/SIMON

SECURITE SOCIALE ACCIDENT DU TRAVAIL/FAUTE INEXCUSABLE DE L'EMPLOYEUR/MAJORATION DE L'INDEMNITE/PROCEDURE/
EXISTENCE DE LA FAUTE INEXCUSABLE/POUVOIR DE DECISION DE LA CAISSE NON

c question 9

BG phrase caisse sauf ((securite ou assurance) et sociale) et decision

BG SOLUTIONS GROUPE 23

BG et phrase accident et travail

BG SOLUTIONS GROUPE 558

BG SOLUTIONS BLOC 12

BG et phrase faute et inexcusable et (employeur ou patron)

BG SOLUTIONS GROUPE 92

BG SOLUTIONS BLOC 1

BG fin

Figure 37 bis - Formulation d'une question avec la version 4-

Mais on voit de suite que chacun des deux groupes nécessite l'utilisation des localisations de trois des quatre notions, d'où perte de performances. De plus, il n'est pas simple du tout de déterminer quel doit être le contenu des deux groupes alors que ce cas est pourtant relativement peu complexe. Par contre, si on admet la possibilité d'écrire plusieurs niveaux de parenthèses, avec la convention classique que le traitement commence toujours par le contenu des parenthèses les plus internes, le même problème se réduit à un seul groupe :

PHRASE CAISSE SAUF ((SECURITE OU ASSURANCE) ET SOCIALE)

Parvenu à ce stade, on voit donc que les deux problèmes se posent au même niveau, paragraphe ou phrase, mais pas au niveau arrêt. Pour ce dernier, il est donc inutile de changer quoi que ce soit. Les modifications nécessaires pour le passage à la version 4 ne portent que sur le traitement des groupes au niveau paragraphe ou phrase, c'est-à-dire, uniquement sur la section TP.

Du point de vue de l'utilisateur, la nouveauté réside donc dans la possibilité d'écrire des groupes au niveau phrase ou paragraphe comportant plusieurs niveaux de parenthèses, sans aucune restriction sur leur nombre, le traitement commençant par l'intérieur des parenthèses, les opérations s'effectuant de gauche à droite sans priorité en l'absence de parenthèses. Enfin, à l'intérieur comme à l'extérieur des parenthèses, les trois opérateurs ET, OU et SAUF peuvent être utilisés sans restrictions.

Chapitre XV

REALISATION DE LA VERSION 4

UTILISATION D'UN FICHIER "A TROUS"

Afin d'automatiser au maximum le traitement des parenthèses à niveau multiple, il est procédé en premier à la traduction du groupe en notation polonaise. Il s'agit ici de la transposition dans ce domaine d'une technique classique en compilation. Cette notation consiste essentiellement à ne pas écrire les opérateurs entre les opérandes, mais derrière. Cela revient donc à écrire par exemple :

SECURITE SOCIALE ET

au lieu de :

SECURITE ET SOCIALE

Si on regarde de plus près, on peut remarquer que cette façon de présenter une opération est plus logique car plus conforme à l'enchaînement des différentes phases du traitement. Il faut en effet, appréhender les deux opérandes avant de commencer l'opération proprement dite et l'ordre dans lequel on utilise les informations est bien : 1er opérande , 2ème opérande, opérateur.

Si on doit traiter une expression importante, on utilise une pile pour les résultats intermédiaires. Le traitement

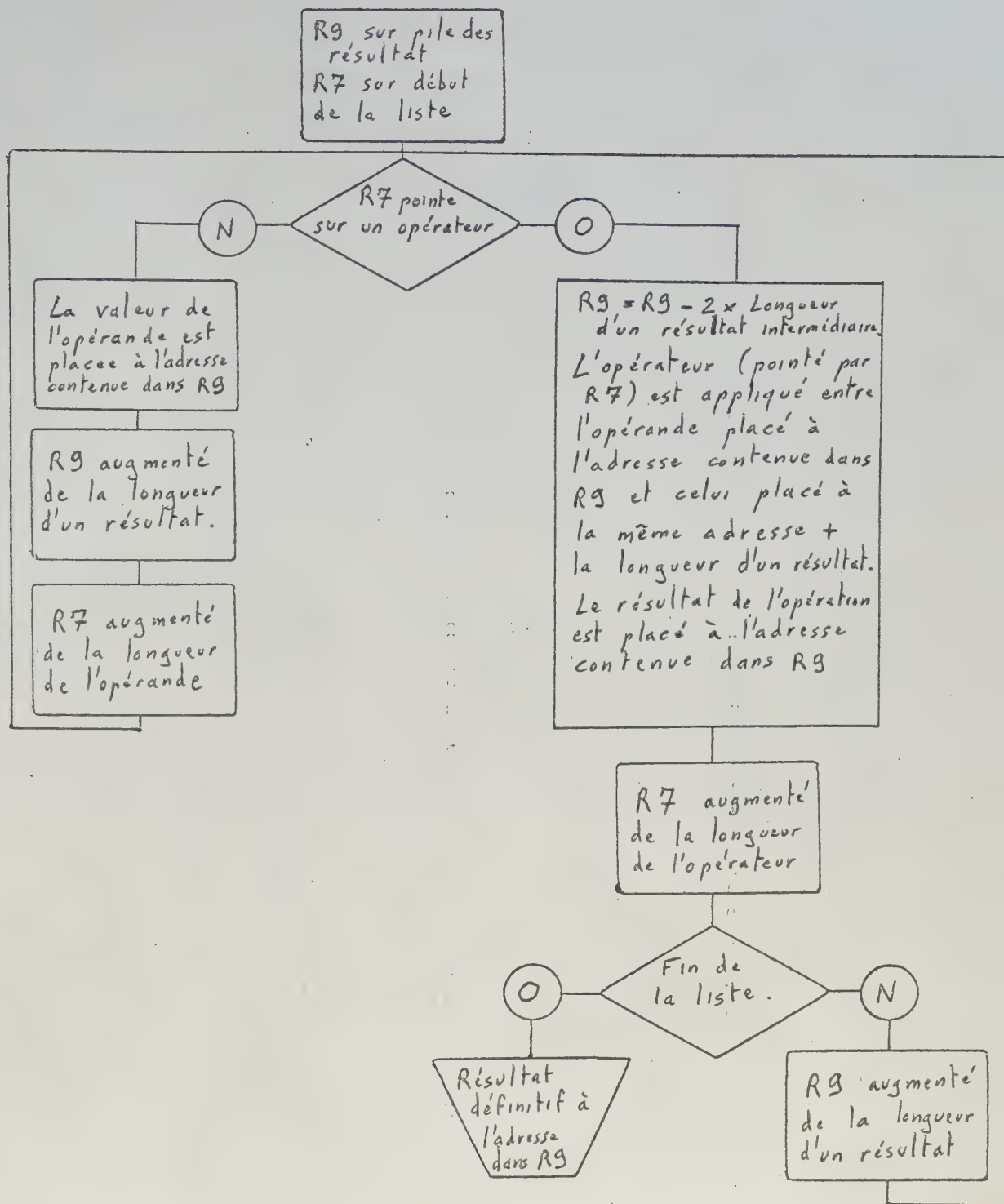


Figure 35 - PRINCIPE DU TRAITEMENT D'UNE EXPRESSION PLACÉE EN NOTATION POLONAISE DANS UNE LISTE.

s'effectue de gauche à droite. Si on rencontre un opérande on le place dans la pile, par contre si on rencontre un opérateur, on l'applique entre les deux éléments du dessus de la pile, le résultat prenant alors la place du premier, le second étant supprimé (voir fig. 35). Le traitement étant fini, il doit rester un seul élément dans la pile, qui est le résultat définitif.

Ainsi :

A B C OU ET donnera successivement :

			C	
	B	B	(B OU C)	
A	A	A	A	(A ET (B OU C))

Le premier travail à effectuer est donc le passage du groupe en notation polonaise. Ceci s'effectue très facilement en utilisant encore une pile et une liste (voir fig. 36). On traite le groupe d'origine de gauche à droite. Si on trouve un opérande (ici une notion), on le place dans la liste. Si on trouve un opérateur, on le place sur la pile après avoir placé dans la liste l'opérateur précédemment présent sur le dessus de la pile s'il y en avait un (les conditions seraient un peu plus délicates en cas de priorités relatives des opérateurs entre eux). Si on trouve une parenthèse ouverte, on la place sur la pile, sans rien mettre dans la liste. Si on trouve une parenthèse fermée, on vide la pile dans la liste jusqu'à la rencontre d'une parenthèse ouverte qu'on supprime de la pile. Lorsqu'on atteint l'extrémité du groupe d'origine, on vide dans la liste tous les opérateurs restant dans la pile.

11111S((22222033333)E44444)E55555

111112222233333044444E55555E

11111

22222

11111

33333

22222

11111

(22222033333)

11111

44444

(22222033333)

11111

((22222033333)E44444)

11111

(11111S((22222033333)E44444))

55555

(11111S((22222033333)E44444))

((11111S((22222033333)E44444))E55555)

Figure 37 - Traitement d'un groupe après sa transcription en notation polonaise.

On trouvera sur la figure 37 un exemple de groupe assez complexe, tel qu'il se présenterait à l'entrée dans la partie du programme de traitement des groupes au niveau paragraphe ou phrase. Chaque notion est représentée par un nombre de 5 chiffres et chaque opérateur par une lettre. Sur la seconde ligne, on a le même groupe écrit en notation polonaise. La suite de la figure donne une image de la pile des résultats intermédiaires après le traitement de chaque élément de l'expression (notion ou opérateur).

Dans tout ce qui précède, nous avons présenté la pile des résultats intermédiaires comme contenant une représentation formelle. En fait, il faut bien voir qu'il n'en est rien, chaque élément de cette pile devant contenir une valeur. Et dans notre cas, une valeur c'est un ensemble de localisations. Mais alors il devient évident que ces éléments vont être de taille variable, puisque le nombre de localisations de chaque notion qui intervient n'est pas constant. On va placer cette pile sur disque, chaque élément étant constitué par un ou plusieurs enregistrements.

Ce nouveau fichier est organisé suivant le principe des enregistrements "chainés", dit aussi fichier "à trous". Chaque enregistrement du fichier contient dans ses premières positions l'adresse d'un autre. Si on considère le fichier à un moment donné, on y trouve :

1) - Les différents éléments de la pile des résultats intermédiaires. Chaque élément est constitué d'un certain nombre

d'enregistrements, chacun d'eux contenant l'adresse du suivant. En mémoire on trouve une autre pile, constituée par l'adresse du premier enregistrement de chaque élément. L'ensemble des localisations composant un élément de la pile est limité par 5 octets pleins de 1 binaires. Cette limite n'est présente que dans le dernier enregistrement de cet élément.

2) - Des enregistrements actuellement inutilisés qui constituent les "trous". Ces derniers sont également chaînés entre eux, l'adresse du premier trou libre étant conservé en mémoire.

Lorsqu'on doit écrire sur disque un des enregistrements composant un élément de la pile, on utilise le premier trou libre. Mais avant de l'écrire, il faut le lire, ou tout au moins lire son début, pour y prendre l'adresse de l'enregistrement suivant dans la chaîne des trous qui devient maintenant le premier trou libre.

Si on charge en mémoire un élément de la pile, on peut récupérer l'emplacement correspondant sur le disque. Dans ce cas, un enregistrement étant lu, on le réécrit de suite (seulement ses premiers octets) après y avoir placé l'adresse du premier trou libre, l'adresse de cet enregistrement lui-même étant à présent stockée comme celle du premier trou libre.

Cette organisation de fichier procure la plus grande souplesse d'exploitation. En effet, chaque élément de la pile peut nécessiter un nombre quelconque d'enregistrements ; on les

prend dans l'ordre où ils sont chaînés dans la chaîne des trous.. Ces enregistrements à leur tour peuvent être "libérés" dans un ordre quelconque, ils ne seront jamais perdus et de plus sont immédiatement réutilisables. Ceci permet d'avoir un volume total nécessaire qui corresponde exactement au volume maximum qui doive effectivement être stocké à un moment donné. En contrepartie on augmente le nombre d'accès sur disque puisque chaque lecture nécessite une écriture en plus et vice versa. Mais cet inconvénient est largement contrebalancé par la suppression des limitations...

Nous avons choisi comme longueur d'enregistrements la piste entière, ce qui nous a permis de réduire l'adresse de chaînage à 3 octets. En effet l'adresse complète d'un enregistrement est donnée normalement sur 5 octets sous la forme CCHHR. Mais sur les unités à disques, le nombre de cylindres ne dépassant pas 255, le premier octet est toujours à zéro. Comme ici, nous n'avons qu'un enregistrement par piste, le dernier octet qui est le N° de l'enregistrement dans la piste est donc toujours à 1. Nous avons donc réduit l'adresse de chaînage à CHH. De plus, l'adresse du premier trou libre est conservée dans le premier enregistrement du fichier, c'est-à-dire, sur la piste 0 de son premier cylindre. Au début du traitement d'un groupe au niveau paragraphe ou phrase, cette adresse est lue. Pendant tout le traitement, elle est maintenue à jour en mémoire et elle n'est réécrite sur disque qu'à la fin du traitement du groupe.

Le traitement d'un groupe nécessite trois zones d'entrée-sortie correspondant au fichier à trous : deux d'entrée correspondant à la lecture de deux éléments de la pile des résultats intermédiaires et une de sortie pour le résultat de l'opération. Il suffit donc de trois zones de 3625 octets. Les deux premières sont placées dans la première zone de 10000 octets et la troisième dans la zone 3. La zone 2 recevra à la fin du traitement la solution du groupe sous la forme d'un bit par arrêt.

Le traitement proprement dit du fichier à trous comprend six routines :

1) - Attribution d'un premier enregistrement à un élément de la pile. C'est le premier trou libre. Une lecture est de suite faite pour récupérer l'adresse du trou libre suivant.

2) - ECZC : Ecriture d'un enregistrement avec blocage du trou libre suivant. Cette routine est utilisée dans le cas où ce n'est pas le dernier pour cet élément. On est donc certain qu'il y en aura encore un à écrire et on le bloque de suite.

3) - ECZCA : Elle est analogue à la précédente mais sans blocage d'un autre trou. On l'utilise pour l'écriture du dernier enregistrement de chaque élément.

4) - LDZA : C'est la lecture d'un enregistrement dans la première des trois zones d'entrée-sortie , mais sans libération du trou. On va l'utiliser pour la lecture du premier enregistrement de l'avant dernier élément de la pile avant sa combi-

naison avec le dernier à l'aide d'un opérateur. Pourquoi, dans ce cas, ne pas libérer l'emplacement ? Tout simplement parce qu'il va falloir un emplacement pour le résultat et que ce dernier doit remplacer dans la pile l'avant dernier élément. On va donc attribuer au résultat le même premier enregistrement qu'à cet élément, ce qui permet de ne rien toucher à la pile des adresses qui est en mémoire.

5) - LDZAT : C'est encore une lecture dans la même zone. Mais cette fois, l'emplacement sur disque est libéré. On va l'utiliser pour la lecture de tous les enregistrements suivants de l'avant dernier élément de la pile, ainsi que pour la lecture de tous les renseignements du résultat final pour placer la solution dans la 2ème zone de 10000 octets.

6) - LDZB : Lecture d'un enregistrement dans la 2ème zone d'entrée sortie avec libération du trou. On s'en sert pour toute lecture du dernier élément de la pile lorsqu'il doit être combiné avant l'avant dernier.

Pour toutes ces routines, on utilise encore l'accès physique sur disque. Le programme canal utilisé est celui déjà existant servant pour les autres fichiers, seul le dernier CCW change suivant la routine. Lorsqu'on désire lire ou écrire uniquement l'adresse de chainage, la longueur n'est plus que de trois octets afin de gagner du temps, l'adresse étant en tête d'enregistrement .

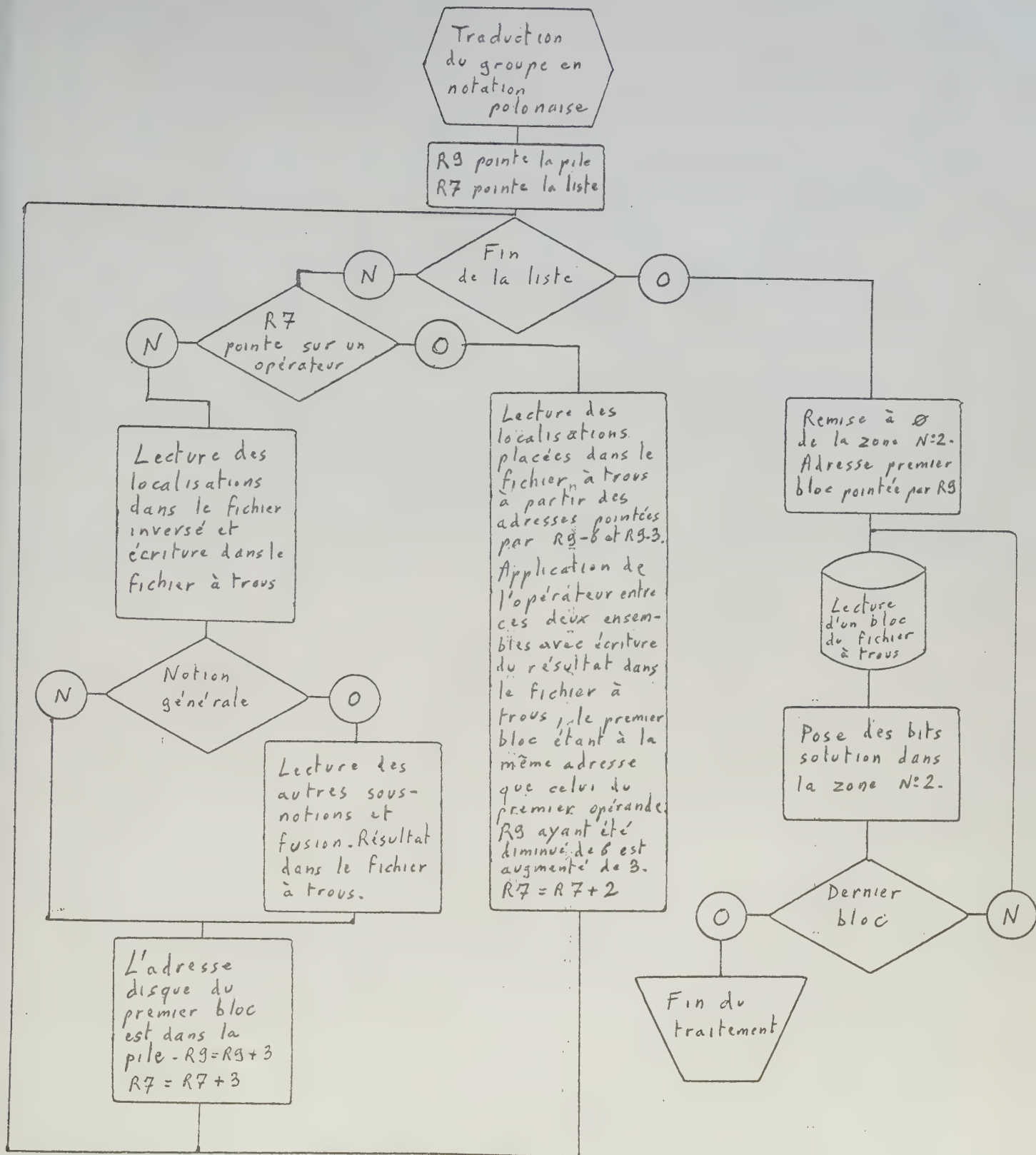


Figure 38 - ORGANIGRAMME GENERAL DU TRAITEMENT D'UN GROUPE AU NIVEAU PARAGRAPHE OU PHRASE (à partir de la version 4).

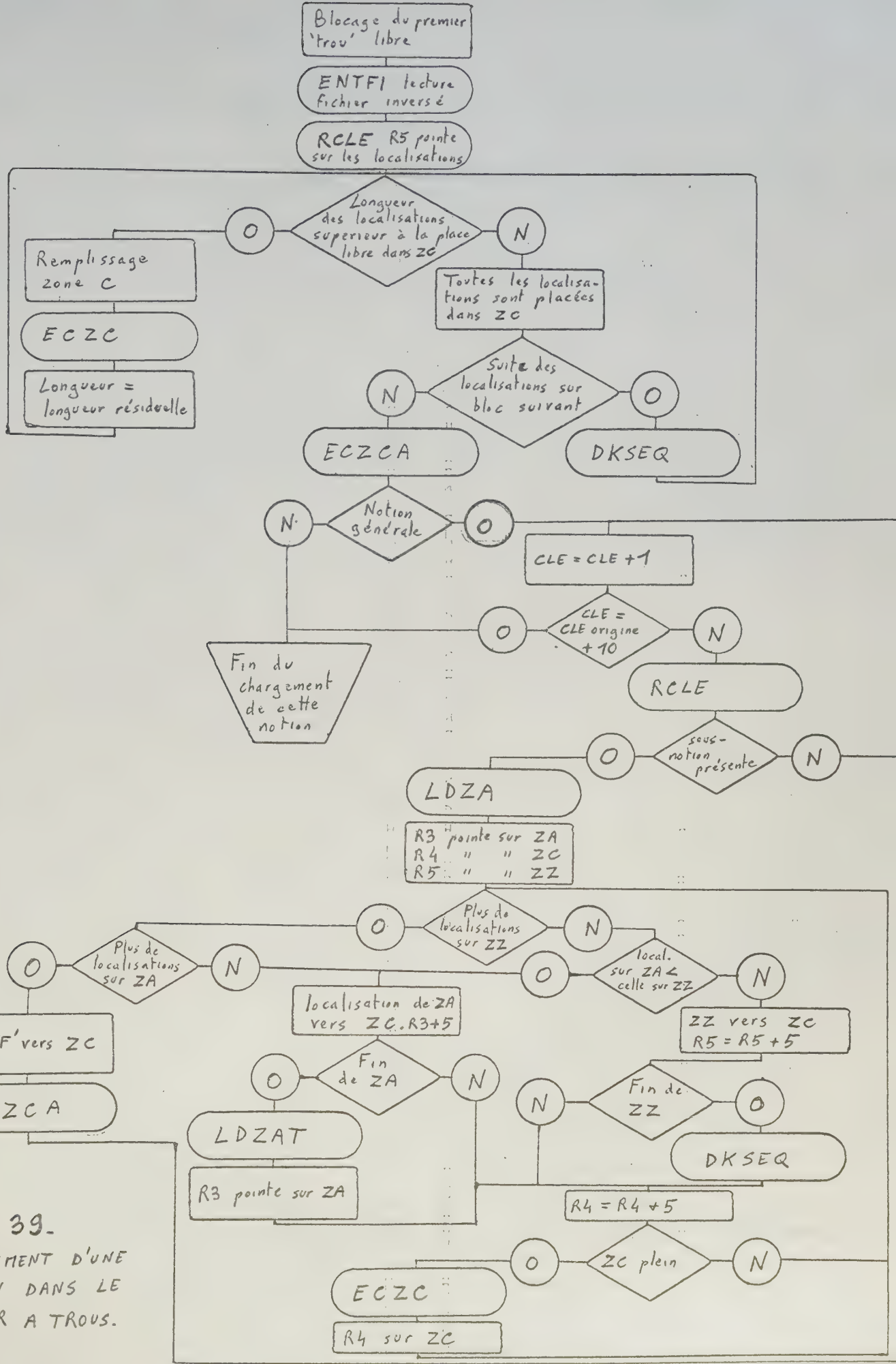


Figure 39.
CHARGEMENT D'UNE
NOTION DANS LE
FICHIER A TROUS.

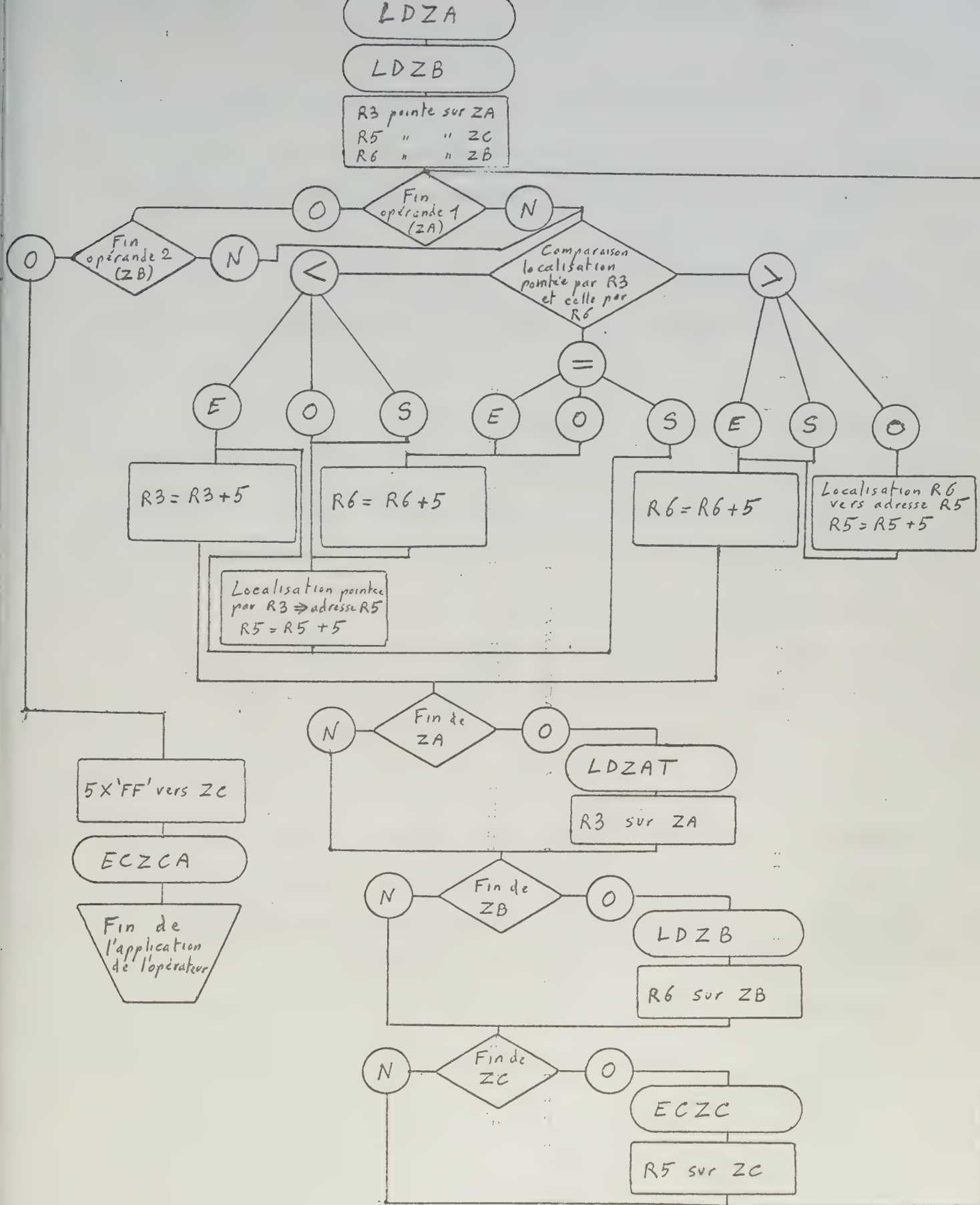


Figure 40 - APPLICATION D'UN OPERATEUR
(section TP à partir de la version 4).

Si une notion est une notion générale, on commence par placer sur la pile les localisations de la première sous-notion. Ces dernières sont ensuite rechargées en mémoire dans la première zone d'entrée-sortie (routines LDZA et LDZAT) et fusionnées avec celles de la sous-notion suivante, le résultat étant écrit sur disque (routines ECZC et ECZCA). On recommence jusqu'à épuisement des sous-notions.

On trouvera sur les figures 38, 39 et 40 les organigrammes correspondant au traitement des groupes au niveau paragraphe et phrase. On notera que ce traitement se subdivise lui-même en quatre parties bien distinctes :

- 1) - Transformation du groupe en notation polonaise.
- 2) - Mise dans la pile des localisations d'une notion.
- 3) - Opération entre deux éléments de la pile.
- 4) - Mise en place du résultat définitif.

Il nous faut enfin signaler que le fichier à trous est préparé par un programme indépendant (D14) qui se contente d'établir un premier chainage en séquence. Tant que la portion de disque correspondante n'a pas été utilisée pour un autre travail, il n'est pas nécessaire de réutiliser ce programme puisqu'à l'issue du traitement de tout groupe tous les enregistrements forment à nouveau une chaîne continue.

Chapitre XVI

LA VERSION 5 DU PROGRAMME DE RECHERCHE

LA POST-SELECTION

En dépit de tous ses perfectionnements, la version 4 présentait encore une lacune, celle de ne pas permettre une sélection sur une plage de valeurs numériques. A vrai dire, ceci n'est pas très important si on n'interroge que sur le "fond", ce qui est le cas avec les fichiers de la Cour de Cassation. Mais si on veut passer au "fait", ce qui est le cas avec les Cours d'Appel, on se heurte inévitablement à ce problème. En effet, on va alors trouver dans le fichier des renseignements numériques tels que le nombre d'enfants, le montant des revenus ou celui de l'indemnité, etc...

Notre système permet déjà en version 4 d'interroger sur de telles valeurs, soit que les nombres soient de moins de 5 chiffres et constituent donc leur propre numéro de notion, soit que l'on ait attribué un N° à chacun de ces nombres. Mais il ne permet l'interrogation que par présence ou absence. Si on désire la présence d'un nombre compris entre deux valeurs, il n'y a pas d'autre solution que de constituer un sous-groupe avec tous les N° de notion possibles dans cet intervalle. Il est bien évident que ce procédé est très lourd et même pratiquement inutilisable dans le cas de plages étendues.

Une première solution aurait consisté en la génération automatique des sous-groupes. Cette solution pourrait à la rigueur être utilisée dans le cas de valeurs inférieures à 10.000 puisque dans ce cas les N° de notion, et par suite les enregistrements du fichier inversé, se suivent. Par contre, si les valeurs sont plus grandes, il faudrait commencer par un "balayage" d'une partie du fichier lexique pour récupérer les N° de notion et ensuite faire autant d'accès complets au fichier inversé qu'il y a de notions dans la plage considérée. Enfin, et c'est peut-être le plus grave, cette méthode oblige à donner un N° de notion à toutes les formes numériques de 5 chiffres et plus. Or, dans la plupart des cas, ce ne sera jamais la valeur exacte qui présentera de l'intérêt. C'est ainsi que la connaissance des revenus annuels d'un individu à un franc près est inutile, mais qu'il est pourtant difficile de fixer arbitrairement des tranches....

Puisqu'il est difficile d'effectuer la sélection sur le fichier inversé, il reste à examiner la possibilité de le faire sur le fichier direct, c'est-à-dire, celui des textes. Il est bien sûr impensable d'examiner la totalité du fichier des textes et c'est ainsi qu'est apparue l'idée de "post-sélection". En effet, une question ne va jamais porter uniquement sur des valeurs numériques. On peut donc commencer par interroger de la façon habituelle à l'aide des autres critères. Ce premier stade donne un certain nombre de solutions et donc de textes qui pourraient être édités. Il suffit alors d'examiner

ces textes en utilisant les conditions numériques et de n'éditer que ceux qui remplissent également ces dernières.

En fait, une condition numérique n'est jamais seule, c'est-à-dire qu'on ne cherche pas uniquement un nombre situé dans telle plage de valeurs mais on veut en plus que ce nombre soit un revenu ou une indemnité ou encore tout autre chose... On peut aussi désirer sélectionner uniquement les arrêts qui présentent deux conditions numériques distinctes... Tout ceci impose donc de pouvoir associer une valeur numérique avec autre chose. Cette association, nous l'avons prévue au niveau phrase, c'est-à-dire, que nous chercherons les phrases contenant à la fois une valeur numérique située dans telle plage et un ou plusieurs mots destinés à caractériser cette valeur.

Ces termes associés peuvent s'indiquer de deux façons :

1) - Une forme- α quelconque, par exemple REVENU.

Dans ce cas il faudra trouver dans une des phrases du texte le mot exact REVENU tel qu'il est écrit ici, c'est-à-dire que le même mot au pluriel n'est pas valable.

2) - Une racine de longueur quelconque suivie d'un point, par exemple REVENU. . Dans ce cas tout terme commençant par ces six mêmes lettres sera considéré comme valable, et donc aussi bien le singulier que le pluriel.

Les conditions numériques vont s'indiquer par NN (qui symbolise numérique) suivi de une ou deux des expressions sui-

vantes :

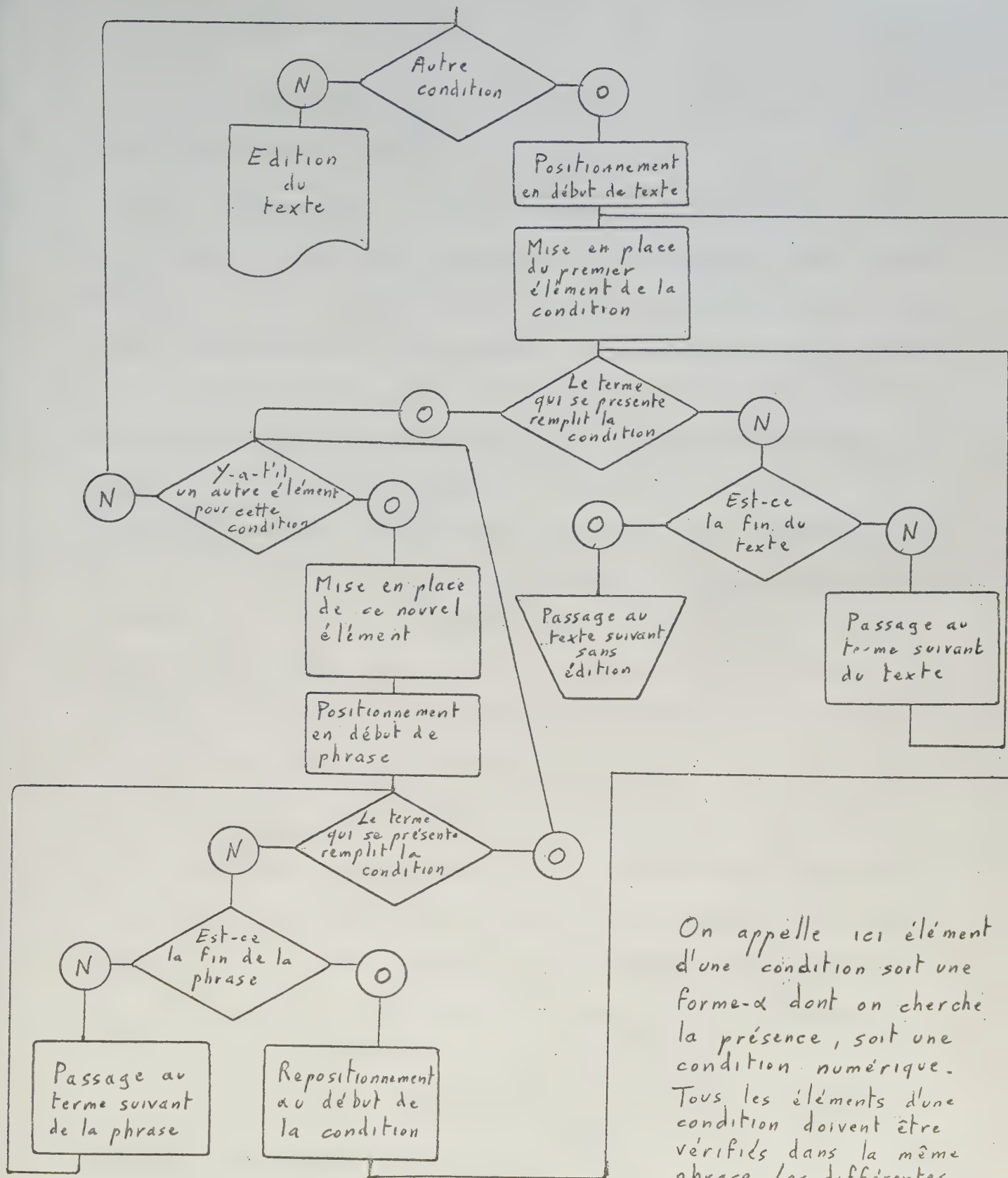
EG	suivi	d'une	valeur	pour	demandeur	l'égalité	
SUP	"	"	"	"	"	plus grand	
DE	"	"	"	"	"	"	ou égal
INF	"	"	"	"	"	inférieur	
A	"	"	"	"	"	"	ou égal

C'est ainsi que l'expression NN DE 1853 A 2547 sélectionnera tout nombre compris entre 1853 et 2547, bornes comprises.

Les conditions nécessaires à la post-sélection sont introduites dans l'instruction de fin de question. Le mot FIN doit être suivi d'un astérisque lui-même suivi des critères. On peut demander la présence simultanée dans la même phrase de plusieurs termes et de plusieurs valeurs numériques. Dans ce cas, on pourra placer un ET entre les différents éléments. Cet opérateur est pour l'instant sans effet puisque c'est le seul possible, mais il a été introduit en prévision d'une extension du système avec les deux autres opérateurs. Exemple : FIN REVENUE. ET ANNUEL. ET NN SUP 25000.

Si on désire utiliser plusieurs conditions simultanées sur des phrases différentes, il faut les écrire les unes derrière les autres en les séparant par des / . Par exemple : FIN REVENUE ET ANNUEL ET NN SUP 25000 / INDEMNITE ET NN INF 3000 donnera l'édition des seuls textes présentant à la fois :

- Une phrase contenant les deux mots revenu et annuel ainsi qu'un nombre supérieur à 25000.



On appelle ici élément d'une condition soit une forme- α dont on cherche la présence, soit une condition numérique. Tous les éléments d'une condition doivent être vérifiés dans la même phrase. Les différentes conditions peuvent s'appliquer à des phrases différentes du même texte.

Figure 41. Principe de la 'post-sélection'.

- Une autre phrase contenant le mot indemnité et un nombre inférieur à 3000.

Réalisation : Elle est obtenue par une modification de la section TXT. Cette modification a été placée après le moment où le texte a été chargé en mémoire et préparé pour l'édition. Les conditions sont examinées de gauche à droite de même que le texte. Les formes- α , complétées à droite par des blancs, sont comparées sur 25 caractères ou sur une longueur égale au nombre de caractères précédant le point dans le cas d'une sélection par racine. Les valeurs numériques sont comparées, après passage en décimal condensé sur 8 octets, soit un maximum de 15 chiffres, avec des bornes inférieures et supérieures, les deux bornes étant égales dans le cas où on a demandé EG. Seules les valeurs positives sont acceptées. On trouvera sur la figure 41 l'organigramme de cette recherche.

Utilisation : Le "langage" utilisé est assez simple et ne pose guère de problème. Il y a toutefois une difficulté venant des différents termes utilisés pour caractériser la même chose. Il faudrait systématiser la rédaction des abstracts de façon à être sûr de pouvoir retrouver ce qu'on cherche. Au point de vue performance on peut dire qu'en gros :

- L'examen d'un texte provoquant sa "non-édition" est plus rapide que l'édition elle-même.

- L'examen , suivi de l'édition est évidemment un peu plus long. Mais sur l'ensemble d'une question, on a sen-

C QUESTION AVEC POST-SELECTION

REPONSE FINALE

***FIN*ARTICLES ET NN DE 1380 A 1383

65 1 C378 11/06/65

CIE D ASSURANCES FRANKFURTER VERSICHERUNG GESELLSCHAFT/MICKEL

RESPONSABILITE CIVILE/CHOSES INANIMEES/ARTICLES 1382 ET 1384 DU CODE CIVIL/ACTION DECLAREE RECEVABLE SUR LE FONDEMENT DE L ARTICLE 1384/PRISE EN CONSIDERATION DE LA FAUTE DU GARDIEN/MOTIF SURABONDANT

RESPONSABILITE CIVILE/CHOSES INANIMEES/ARTICLE 1384 DU CODE CIVIL/RESPONSABILITE DE PLEIN DROIT/EXONERATION/ CARACTERE IMPREVISIBLE ET INEVITABLE/CIRCULATION ROUTIERE/PIETON/PRESENCE SUR LA CHAUSSEE/AUTOMOBILISTE RECONNAISSANT

L AVOIR APERCU

ASSURANCE RESPONSABILITE/GARANTIE/LIMITATION FIXEE PAR LA POLICE/PERMIS DE CONDUIRE REGULIER/ACCIDENT CAUSE EN FRANCE PAR UN TOURISTE ETRANGER

c question avec post-selection

BG accident et circulation

BG SOLUTIONS GROUPE 28

BG et piéton

BG SOLUTIONS GROUPE 73

BG SOLUTIONS BLOC 5

BG fin*articles et nn de 1380 a 1383

BG

Figure 42 - Post-selection avec 'champ numérique'

C QUESTION AVEC RECHERCHE SUR LES PARTIES

REPONSE FINALE ***FIN*DELEBOIS/FLANDRE

66 2 0509 29/04/66

COMPAGNIE LA FLANDRE ET AUTRE/VEUVE DELEBOIS ET AUTRES

RESPONSABILITE CIVILE/CHOSES INANIMEES/ARTICLE 1384 DU CODE CIVIL/RESPONSABILITE DE PLEIN DROIT/EXONERATION/
FAUTE EXCLUSIVE DE LA VICTIME/CIRCULATION ROUTIERE/PIETON/TRAVERSEE DE LA CHAUSSEE/TRAVERSEE DANS UN PASSAGE
CLOUTE/POSITION REGULIERE A L'INTERIEUR DU PASSAGE NON

CASSATION/MOYEN NOUVEAU/SECURITE SOCIALE/ACCIDENT DU TRAVAIL/TIERS RESPONSABLE/RECOURS DES CAISSES ET DE
LA VICTIME/APPEL/AUGMENTATION DES REMBOURSEMENTS ACCORDES AUX CAISSES/MAINTIEN DU PREJUDICE GLOBAL ET DE L
INDEMNITE COMPLEMENTAIRE DE LA VICTIME

c question avec recherche sur les parties

BG accident et circulation

BG SOLUTIONS GROUPE 28

BG et pleton

BG SOLUTIONS GROUPE 73

BG SOLUTIONS BLOC 5

BG finedelebois/flandre

BG

*Figure 42 bis. Interrogation avec utilisation
des noms des parties.*

siblement le même temps que celui qui serait nécessaire pour l'édition complète de tous les textes sélectionnés au premier stade.

Enfin, l'examen portant sur tout le texte permet de sélectionner sur des éléments absents du fichier inversé, tels le nom des parties, la date ou des formes numériques absentes du lexique.

-o-o-o-o-o-

Chapitre XVII

RESULTATS OBTENUS

Si on cherche à faire une sorte de bilan de cette expérience, il est nécessaire de le faire de plusieurs points de vue. C'est en effet le propre de tout travail pluri-disciplinaire d'avoir des répercussions dans chacune des branches.

1) - Côté informatique : Tous les programmes intervenant dans l'ensemble du système fonctionnent bien, certains d'entre eux ayant un rendement particulièrement élevé. C'est le cas du plus important, le programme de recherche lui-même. Compte tenu des multiples possibilités qu'il offre, son encombrement mémoire est très faible. Sous sa forme actuelle, il occupe dans son ensemble 54 K. Mais il serait très possible de réduire cette taille car il n'y a en fait que 14 K de programme, le reste étant constitué par les 4 zones de 10000 octets. La zone n° 4 peut très facilement être remplacée par 3 pistes sur un disque. Des trois autres zones, on pourrait encore en retirer une sans avoir trop de difficulté. Dans ces conditions l'ensemble du programme de recherche doit pouvoir tenir dans une partition de 34 K.

Ses performances en vitesse d'exécution sont également très bonnes. C'est ainsi que sur un fichier de 12000 ar-

Le groupe traité est:

PHRASE CAISSE SAUF ((SECURITE OU ASSURANCE) ET SOCIALE)

Au départ, le fichier à trous, supposé limité à 10 enregistrements numérotés de 1 à 10, vient d'être initialisé par le programme D14, le chainage étant donc en séquence.

L'essai a été effectué avec un fichier portant sur 4 ans (1965 à 1968).

Le nombre de localisations est sensiblement de:

- 600 pour le mot CAISSE.
- 2500 pour SECURITE.
- 1000 pour ASSURANCE.
- 2500 pour SOCIALE.

Dans le fichier inversé, on doit avoir 4 blocs pour CAISSE, 20 blocs pour SECURITE, 7 pour ASSURANCE et 20 pour SOCIALE.

Le seul accès au fichier inversé nécessite donc:

- 4 appels au sous-programme ENTFI, chacun d'eux provoquant 3 accès disque (2 pour les index et 1 pour lire un bloc), soit au total 12 accès disque.

- Autant d'appels au sous-programme DKSEQ qu'il y a de blocs supplémentaires, soit $3+19+6+19=47$. A chaque appel de ce sous-programme on a un accès disque, ce qui fait donc 47 accès de plus.

L'utilisation du fichier à trous va nécessiter en plus:

- Un accès disque pour chaque notion (chainage des trous), soit donc 4 accès.

- 2 accès à chaque appel de ECZC, appelé 13 fois, soit 26.
- 1 accès à chaque appel de ECZCA appelé 7 fois.
- 1 accès à chaque appel de LDZA appelé 3 fois.
- 2 accès à chaque appel de LDZAT appelé 7 fois, soit 14.
- 2 accès à chaque appel de LDZB appelé 10 fois, soit 20.

On a donc au total 74 accès pour le fichier à trous.

Nombre total d'accès disque : $12+47+74=133$.

Fonction	Sous-programme	Bloc	Chaine des "trous"	Pile des résultats intermédiaires.
Chargement CAISSE	ECZCA	1	2.3.4.5.6.7.8...	1
Chargement de SECURITE	ECZC	2	4.5.6.7.8.9.10	2.3.4.5 1
	ECZC	3	5.6.7.8.9.10	
	ECZC	4	6.7.8.9.10	
	ECZCA	5	6.7.8.9.10	
Chargement de ASSURANCE	ECZC	6	8.9.10	6.7
	ECZCA	7	8.9.10	2.3.4.5 1
Calcul de (SECURITE OU ASSURANCE)	LDZA	2	8.9.10	2.6.3.4 1
	LDZB	6	6.8.9.10	
	ECZC	2	8.9.10.10	
	LDZAT	3	3.8.9.10.10	
	ECZC	6	8.9.10	
	LDZAT	4	4.8.9.10	
	ECZC	3	8.9.10	
	LDZB	7	7.8.9.10	
	LDZAT	5	5.7.8.9.10	
	ECZCA	4	5.7.8.9.10	
Chargement de SOCIALE	ECZC	5	8.9.10	5.7.8.9 2.6.3.4 1
	ECZC	7	9.10	
	ECZC	8	10	
	ECZCA	9	10	
Calcul de ((SECURITE OU ASSURANCE)ET SOCIALE)	LDZA	2	10	2.7.8.9 1
	LDZB	5	5.10	
	LDZAT	6	6.5.10	
	LDZB	7	7.6.5.10	
	ECZC	2	6.5.10	
	LDZAT	3	3.6.5.10	
	LDZB	8	8.3.6.5.10	
	ECZC	7	3.6.5.10	
	LDZAT	4	4.3.6.5.10	
	LDZB	9	9.4.3.6.5.10	
	ECZC	8	4.3.6.5.10	
	ECZCA	9	4.3.6.5.10	
Calcul de l'expression complète	LDZA	1	4.3.6.5.10	1
	LDZB	2	2.4.3.6.5.10	
	LDZB	7	7.2.4.3.6.5.10	
	LDZB	8	8.7.2.4.3.6.5.10	
	LDZB	9	9.8.7.2.4.3.6.5...	
	ECZCA	1	9.8.7.2.4.3.6....	
Exploitation	LDZAT	1	1.9.8.7.2.4.3.6.5.10	

Figure 44.UTILISATION DU FICHIER A TROUS POUR LE GROUPE DE LA Fig.43.

rêts, on arrive à un temps de 3 à 5 secondes pour une question posée par cartes, composée de 2 ou 3 groupes au niveau arrêt avec 2 ou 3 notions par groupe. Ce temps comprend l'édition du résultat, pour une solution comportant 2 à 3 arrêts. Si on interroge au niveau phrase ou paragraphe avec la version 4, les temps augmentent dans de notables proportions. C'est ainsi que sur les mêmes fichiers, l'obtention de la solution du groupe :

PHRASE CAISSE SAUF ((SECURITE OU ASSURANCE) ET SOCIALE :
dont on trouvera le détail du traitement sur les figures 43 et 44, nécessite 11 secondes. Mais il faut noter que cet exemple est un peu un extrême étant donné que toutes les notions qui interviennent sont très fréquentes et possèdent entre 2000 et 3000 localisations ce qui nécessite 145 accès aux disques (133 + 12 pour le lexique). On peut donc dire qu'une question "normale" ne nécessite jamais plus d'une minute d'ordinateur. Ce temps peut être augmenté dans le cas d'une édition de très nombreux arrêts. C'est le cas des "bilans de jurisprudence" pour lesquels la question se réduit souvent à une seule notion assez fréquente.

Mais l'intérêt de cette expérience ne s'est pas limité aux seuls programmes de ce système. Il y a eu également des "retombées" dans des domaines autres que l'informatique juridique. C'est ainsi que les premiers programmes de la chaîne ont pu être utilisés pour des études de vocabulaire sur des textes politiques anglais du 18ème siècle. Mais c'est surtout la mise au point de

méthodes d'accès sur disque qui a eu le plus de répercussion. L'organisation du fichier de recherche lui-même offre le séquentiel indexé en longueur variables sous DOS. Cet avantage n'a pas échappé aux spécialistes qui en ont eu connaissance et certains l'ont utilisé dans des domaines divers. Quant à l'utilisation des fichiers "à trous", elle est déjà prévue pour la constitution d'un fichier de dossier médical pour le Centre Anti-Cancéreux de Montpellier.

2) - Côté juridique : Malgré toute la souplesse du système, malgré les avantages de l'aspect conversationnel, il ne semble tout de même pas possible d'envisager dans un proche avenir la constitution d'un fichier central relié à des terminaux situés directement chez les praticiens. Il y a deux raisons à cela :

a) - Il faut un certain entraînement et une certaine formation pour parvenir à interroger correctement. Il ne suffit pas d'avoir parfaitement bien saisi le problème juridique posé pour pouvoir obtenir des solutions correctes. Il est nécessaire de manier correctement les trois opérateurs logiques, d'user avec à propos des trois niveaux d'interrogation, etc... Il ressort très nettement de notre expérience que tout ceci ne s'acquiert pas en un jour et que certains même, excellents juristes par ailleurs, y sont parfaitement réfractaires. Il sera donc vraisemblablement préférable d'avoir des centres de documentation équipés de terminaux, ces derniers étant utilisés par des juristes spécialement formés pour cela. De tels centres ser-

viraient donc d'intermédiaires entre les praticiens et l'ordinateur.

b) - Avant de pouvoir passer à une exploitation en vraie grandeur il faudrait être certain que les documents de base entrés dans le système soient parfaitement adéquats. Or, il semble bien que ceux que nous avons utilisés, les titres principaux du bulletin, soient pour le moins imparfaits pour cet usage. Les diverses expériences ont montré qu'on ne pouvait guère attendre un taux de réponse correcte de plus de 50 %. Ce n'est certes pas suffisant. Mais ceci a été très intéressant en ce sens qu'il nous a obligé à étudier qu'elle devait être la structure du document de base. Une première série de règles a été dégagée. En appliquant ces règles à un certain nombre d'arrêts, il a été possible de fabriquer une série "d'abstracts" qui ont été soumis au même traitement. Les résultats ont été encourageants puisqu'une augmentation du taux de réponse de plus de 10 % a été obtenue dès le premier essai. Actuellement cette étude se poursuit en liaison avec d'autres équipes, en particulier celle du Professeur BERTRAND d'AIX-EN-PROVENCE et celle de Monsieur DURIEUX des EDITIONS TECHNIQUES.

Les résultats acquis dans le domaine juridiques, les perspectives d'avenir et la nature des travaux en cours ne seront pas exposés ici plus en détail. En effet, ils l'ont été déjà dans plusieurs rapports dont on trouvera la liste en annexe. Nous signalerons tout de même l'intérêt suscité par notre expérience dans les milieux juridiques, non seulement français, mais encore

de quelques pays européens (Luxembourg, Suisse, Espagne, Belgique, Italie, Danemark, etc...) et américains (Canada, Uruguay, Brésil et République Argentine). Beaucoup de personnes intéressées ont effectué le déplacement à Montpellier, quelques unes ayant même effectué un véritable stage.

Enfin, tout en restant dans le domaine de la documentation automatique, il ne faut pas oublier de regarder hors du domaine juridique. Il est en effet évident que les techniques mises au point n'ont rien de spécifiquement juridique. Il est fort probable que certaines options ont été prises en fonction de ce domaine d'application particulier et que dans d'autres circonstances, nous aurions pris les choses d'une façon différente. Mais enfin le problème de documentation automatique présente toujours des aspects identiques dans les différents champs d'application. Aussi il nous semble possible d'utiliser notre système pour d'autres problèmes. Et c'est ainsi que nous venons d'entreprendre un essai sur une banque de données relative à des logements. Les documents de base contiennent les caractéristiques des logements, l'interrogation permettant de sélectionner tous les appartements ou maisons individuelles possédant certaines caractéristiques. Cette expérience qui promet d'être d'un grand intérêt, est effectuée en liaison avec le C.E.T.E. d'Aix-en-Provence.

Chapitre XVIII

LES DIFFERENTES VOIES

DE LA DOCUMENTATION AUTOMATIQUE

Avant de conclure, nous voudrions essayer de montrer quel pourrait être demain le visage de la documentation automatique à l'aide d'un ordinateur. Ceci ne signifie nullement que nous puissions être certains de l'évolution dans ce domaine. Nous voulons simplement indiquer quelques directions possibles, compte tenu du matériel existant ou annoncé...

La solution que nous avons développée présente certes pas mal d'avantages, mais aussi quelques inconvénients. Tout d'abord, il est évident que son utilisation actuelle, sur le plan du rendement de l'ordinateur, est catastrophique. N'ayant que trois lecteurs de disques 2311 et seulement 64 K de mémoire centrale, nous sommes obligés de passer en mono-programmation et donc de bloquer pour notre programme toutes les ressources de notre installation. Or, en fonctionnement en conversationnel, c'est souvent une immobilisation assez longue qu'il faut envisager. Ceci ne peut être réalisé que la nuit et encore parce que nous n'avons pas assez d'opérateurs pour tourner en permanence. Bien sûr, le passage en "batch" avec interrogation par cartes est toujours possible et est d'un rendement élevé ; mais

plus personne ne veut l'utiliser...

En effet, tout au moins dans les phases d'essai et de recherche, l'apport du conversationnel est prépondérant. Et c'est d'autant plus vrai quand les documents d'origine n'ont pas une structure et un vocabulaire normalisés. On peut alors par tâtonnements successifs approcher de la solution et peu à peu acquérir une technique d'interrogation adaptée aux documents. Il ne faut pourtant pas de faire d'illusion et croire que le mode conversationnel est le seul valable. Nous pensons même qu'il est des cas où il peut être parfaitement inutile et par suite nuisible puisqu'il induit systématiquement une baisse de rendement. Nous verrons un peu plus loin quelles sont les circonstances qui peuvent pousser à cette conclusion.

En fait, le mode conversationnel n'a jamais été prévu pour une utilisation en uni-job. Normalement, c'est à partir d'un terminal que doit se faire l'interrogation. On a alors plusieurs types possibles d'organisation qui peuvent être, en schématisant :

1) - Un seul terminal, accédant au programme de recherche, placé dans une des partitions de la mémoire centrale, d'autres travaux s'effectuant pendant ce temps dans les autres partitions. Cette solution est couteuse sur deux plans :

a) - Mémoire centrale, puisqu'une partition est immobilisée pour l'usage de ce seul terminal qui, d'une part, peut très bien ne pas fonctionner en permanence tout en étant susceptible de le faire à n'importe quel moment et , d'autre part, pré-

sente de nombreux temps morts (transmission des messages, réflexion de l'utilisateur, etc...).

b) - Mémoire auxiliaire, c'est-à-dire, essentiellement des lecteurs de disques. En effet, cela suppose que les fichiers nécessaires à la recherche soient en permanence en ligne et que l'installation dispose d'autres lecteurs pour les programmes travaillant dans les autres partitions. En fait, quels sont les besoins en disques ? Dans notre système actuel, nous avons besoin d'un disque 2311 pour le fichier inversé, le lexique et les zones de travail, ceci pour 5 ans. En fait, il reste beaucoup de place et nous pourrions placer jusqu'à 10 années. Ces dix années représentent environ 30.000 documents à 30 mots en moyenne par document, soit 900.000 mots dont 500.000 possédant un N° de notation. Par contre, les fichiers des textes sont beaucoup plus volumineux et il nous est impossible de placer plus de 5 ans sur le même disque.

En fait, quelle peut être la solution ? Elle ne réside certainement pas dans des disques du type 2311 qui sont de trop faible capacité. Par contre, les disques type 2314, à 30 millions d'octets par pile présentent une bonne solution pour le fichier inversé et le lexique. Dans le cadre d'une grosse installation possédant un lecteur 2314 complet à 9 lecteurs (dont 8 seulement sont utilisables à un instant donné), il n'est nullement impensable d'envisager la réservation permanente d'un , voire de deux, des lecteurs pour la partition de documentation automatique.

Par contre, à notre avis, le support des textes ne devrait pas être un disque, mais plutôt une unité à feuillets magnétiques. La capacité d'une telle unité est de 100 millions de caractères et son accès tout aussi aisé que celui d'un disque. Bien entendu, cet accès est notablement plus lent, mais rien n'empêche d'envisager la recherche sur cette unité comme une "tache" séparée qui serait activée en concurrence avec le programme principal pendant l'édition du texte précédent.

2) - Nous avons signalé plus haut l'existence de temps morts pendant l'interrogation. Il devient alors tentant de les utiliser en faisant partager les ressources entre plusieurs terminaux. Le problème devient alors de parvenir à déterminer le nombre optimum qui doit être branché. Le taux d'utilisation de l'unité centrale par un terminal varie en fonction du type de question (rapport variable entre le temps de recherche et le temps d'édition) et suivant l'utilisateur. L'utilisateur des terminaux, surtout si ce sont des terminaux éloignés (cas d'un réseau national de documentation juridique par exemple) ne sera pas constante et variera suivant les heures, les jours et les époques. Il peut même y avoir des "pointes" et des "creux" parfaitement aléatoires que rien ne peut permettre de prévoir... Le nombre de terminaux possibles varie également en fonction du temps maximum de réponse qui est désiré. Plus ce dernier est court, plus on est obligé de prévoir une surpuissance de l'ordinateur par rapport au nombre de terminaux...

3) - Il y a un danger à pousser trop loin le système évoqué dans le paragraphe précédent, c'est la saturation des ressources de l'ordinateur par la seule partition réservée à la documentation automatique. Si en effet on augmente le nombre de terminaux, cette partition aura toujours quelque chose à faire. Comme d'autre part il semble bien qu'elle doive être prioritaire, il va s'en suivre que peu de chose pourra être effectué en parallèle et en tous cas pas de programme nécessitant beaucoup de traitement en unité centrale. Et d'autre part, si on ne met pas assez de terminaux, on risque de sous-utiliser et les ressources en mémoire centrale et celles en unités périphériques. Mais alors, si ce genre de travail est capable à la limite de saturer un ordinateur, pourquoi ne pas lui en affecter un. En effet, si on y regarde de près, il est inutile d'utiliser un très gros ordinateur. Le traitement en unité centrale n'est pas considérable et de ce point de vue, un 360-40 est déjà surpuissant et quelque chose de l'ordre du 30 suffirait bien. Une taille mémoire de 64 K octets est suffisante pour fonctionner avec plusieurs terminaux, à condition de traiter les choses groupe par groupe. Ceci veut dire que le traitement d'un groupe en provenance d'un terminal étant commencé, on ne pourra prendre en compte les demandes venant d'un autre terminal qu'après achèvement de ce groupe.

Dans cette dernière optique, deux variantes peuvent être envisagées :

a) - Un réseau de terminaux éloignés, les réponses étant éditées sur les terminaux.

b) - Une série de terminaux en mode local (clavier et petit écran cathodique). Dans ce cas seuls pourraient être renvoyés sur le terminal les éléments purement conversationnels actuellement sortis sur la machine à écrire. Les textes ne seraient pas édités sur le terminal, mais les solutions finales seraient envoyées sous forme de références dans un fichier sur disque à gestion dynamique. Ce fichier serait alors exploité en parallèle par un autre petit programme placé dans une 2ème partition (4 à 6 K) qui irait chercher les textes à éditer sur l'unité périphérique ad hoc (si possible sur un autre canal que les disques pour éviter des interférences) et les sortirait sur l'imprimante. Le personnel utilisant les terminaux serait alors une équipe de spécialistes de l'interrogation connaissant bien la technique et le contenu des fichiers. Les questions devraient alors être centralisées sur ce seul centre de traitement, soit par la poste, soit par téléphone, les réponses suivant le même chemin. Il est évident que les délais seraient de la sorte un peu plus long, mais la qualité des réponses risquerait fort d'être meilleure... Quant à la mise à jour des fichiers, elle ne poserait guère de problème puisqu'il suffirait de prévoir dans le planning un arrêt de l'interrogation, éventuellement journalier, qui permettrait de la faire avec toutes les ressources de l'installation. Et il n'est pas impossible que cette solution soit en fin de compte moins onéreuse que le partage d'un gros ordinateur avec une autre exploitation.

Dans tout ce qui précède, nous n'avons pas indiqué quel type de terminal semblait souhaitable, sauf dans le dernier cas. Lorsqu'il s'agit de terminaux éloignés, plusieurs solutions sont offertes, mais à l'heure actuelle, aucune n'est pleinement satisfaisante. Le terminal à écran cathodique allie le silence et une certaine rapidité, mais il ne laisse pas de trace écrite, à moins d'être couplé à un mécanisme de prise de vues. La machine à écrire (le classique télétype) est vraiment trop lent si on désire obtenir dessus l'édition complète des textes. Il y a bien le "terminal lourd" avec imprimante, mais alors l'investissement est important. Il faut toutefois signaler l'apparition depuis peu d'un terminal original, le VIATRON. Ce dernier comprend un clavier, un écran, un enregistreur de bande magnétique (mini-cassette) et un "robot" qui posé sur le clavier d'une vulgaire machine à écrire permet de taper le contenu de la mini-cassette. Il serait alors envisageable de faire apparaître sur l'écran les réponses de type conversationnel et d'enregistrer sur la bande celles correspondant aux textes. Ces derniers pourraient alors être édités en différé, éventuellement sur une deuxième machine, puisque ce terminal fonctionne aussi bien on line qu'off line. Vu en plus le prix modique de ce matériel, il n'est pas impossible que nous puissions un jour essayer cette solution.

Il y a enfin un projet de terminal spécifique pour la documentation automatique (DARIUS, étudié par M. ZANETACCI de la Faculté des Sciences de Paris). Ce terminal est muni d'une mémoire à accès sélectif, mais composée de micro-films. Ces der-

niers contiennent les textes. Ce terminal, recevant les coordonnées du document, est alors capable de le faire apparaître sur un écran. L'utilisateur, l'ayant lu, peut savoir s'il présente effectivement un intérêt et, si c'est le cas, peut en obtenir de suite une reproduction. L'énorme avantage de ce type de matériel serait de permettre de travailler sur des textes relativement gros tout en ayant à transmettre peu de choses sur la ligne (une simple référence) et en supprimant sur l'ordinateur le volumineux fichier des textes. Il y a toutefois deux inconvénients, le premier étant que cet appareil n'est pas encore disponible, le second étant pour la mise à jour des documents. Mais dans le cas de documents tels qu'il ne puisse y avoir que des ajouts (cas de la jurisprudence) ce dernier inconvénient est de peu de poids. C'est peut-être bien là une des solutions d'avenir.

Il y a enfin le cas du traitement de documents à structure parfaitement définie, à contenu normalisé et à types de questions peu nombreux. Dans ce cas qu'apporte le mode conversationnel ? Si on connaît bien la structure des documents, avec un peu d'entraînement, on peut parvenir à poser les questions à coup sûrs. On peut même préparer à l'avance des questions types dans lesquelles il ne reste plus qu'à placer les valeurs désirées. C'est le cas en particulier pour certaines "banques de données" telles que des banques de logements. Les renseignements contenus dans les documents de base correspondent aux caractéristiques des logements (situation, prix, superficie, etc...) et sont donc de nature sensiblement constante. De même, l'interrogation ne

portera que sur la valeur de ces éléments et les questions elles-mêmes auront donc une structure assez fixe. Dans ces conditions, on a tout intérêt à poser les questions par un processus non conversationnel, plus rapide et donc plus rentable. Si on travaille en local, le plus simple est encore de passer les questions sur cartes (ou sur bande magnétique si on possède des enregistreurs sur bande , type MDS par exemple). Dans ces conditions, l'utilisation de terminaux perd une partie de sa justification, sauf si on a absolument besoin d'une réponse quasi instantanée. Si ce n'est pas le cas, une bonne solution peut consister en un enregistrement à bande magnétique placé en un lieu de centralisation des questions relié par ligne P et T à un autre enregistreur situé près de l'ordinateur. Sur vacation, le contenu de la bande est transféré d'un enregistreur sur l'autre et passé ensuite sur l'ordinateur. Les réponses peuvent alors être transmises par la poste ou par téléphone ou télex.

Un dernier point mérite enfin d'être abordé, celui de la mise à jour des fichiers. Le problème est très simple dans le cas de la jurisprudence où un simple ajout périodique de documents est suffisant. Cette mise à jour est effectuée en dehors des moments d'interrogation et en batch-processing. La difficulté est beaucoup plus sérieuse dans le cas de fichier à évolution rapide. C'est ainsi que dans le cas d'une banque de logements, tout logement loué ou vendu doit être supprimé du fichier, de même que tout logement devenant disponible doit y être introduit. Il peut devenir avantageux dans ce cas d'effectuer la mise à jour à une

fréquence plus élevée, voire dans certains cas en permanence et simultanément à l'interrogation. Dans le cas de nombreux changements, la difficulté réside dans les modifications du fichier inversé. Il faut alors prévoir une structure permettant la mise à jour rapide de ce fichier sans être obligé de le restructurer en entier. Une solution possible serait alors de laisser une certaine place dans les blocs de façon à pouvoir y faire des ajouts, une réorganisation étant toutefois nécessaire de temps à autre.

Mais le danger d'une mise à jour "en temps réel" effectuée simultanément à l'interrogation réside dans le risque d'un accès simultané aux mêmes enregistrements par les deux types de travaux. La meilleure solution doit encore consister dans la procédure dite "track hold" qui permet d'interdire l'accès à une piste depuis une autre partition que celle qui l'utilise présentement. De même, il est possible de s'arranger de telle sorte que deux procédures de mise à jour du même fichier ne s'effectuent pas en concurrence...

Il est certain que ces problèmes présentent un certain intérêt, tout au moins du point de vue de la recherche. Il n'est pas certain que ce soit aussi net sur le plan pratique et personnellement il ne nous semble pas. Mais ceci n'a rien d'un jugement définitif...

Chapitre XIX

PERSPECTIVES D'AVENIR

Ce qui vient d'être exposé ne doit pas être considéré comme un résultat définitif, comme la conclusion ultime de notre recherche dans ce domaine. C'est tout au plus une étape. Il reste encore beaucoup à faire et le souhait que nous pouvons former est de voir d'autres informaticiens se joindre à notre équipe. En effet, si le nombre des juristes a sérieusement augmenté, il n'y a encore qu'un seul informaticien, moi-même. Puissent d'autres chercheurs être tentés à la lecture de ce qui va suivre...

Une première direction consisterait en l'amélioration des performances du programme de recherche. Il doit, en effet, être encore possible de l'optimiser sur le plan de l'occupation mémoire et de la rapidité. Il est certain que l'automatisation poussée du processus de traitement des groupes au niveau phrase ou paragraphe mène souvent à des opérations inutiles et à des accès sur disque superflus. C'est ainsi que dans certaines circonstances, des résultats intermédiaires sont écrits sur disque à partir d'une zone pour être immédiatement relus dans une autre ; comme il s'agit du fichier "à trous", cela fait 4 accès disque de trop. Il faudrait également le modifier légèrement

pour pouvoir le faire exécuter dans n'importe quelle partition sans avoir à modifier la valeur de certaines constantes.

Un autre travail, qui va sans doute être entrepris, consistera en la transposition de tous les programmes en vue de leur exploitation sous Opérating System. Il est, en effet, apparu que ces programmes sont susceptibles d'intéresser des utilisateurs de gros systèmes alors qu'au départ nous ne l'avions envisager que pour un système moyen. C'est essentiellement l'utilisation des techniques de multiprogrammation qui les y pousse . En effet, si le système considéré dans son ensemble est gros, chacune des partitions considérée isolément peut être assimilée à un système moyen, voir petit.

Un autre but plus important comme travail serait le passage à un système "télé-processing". Il faudrait alors pousser les choses jusqu'au bout, c'est-à-dire, mettre au point un programme de recherche qui puisse être partagé entre plusieurs terminaux. Ceci nécessiterait la constitution d'une file d'attente des messages et surtout le stockage sur disque des éléments essentiels pour chaque terminal. Dans un premier temps, il pourrait être admis que le programme traiterait chaque groupe sans interruption, le passage d'un terminal à un autre se faisant entre les groupes. Si d'autre part on admet qu'il n'y a pas de priorité relative des terminaux et que les groupes sont traités dans l'ordre où ils ont été soumis, on ne doit pas se heurter à de trop grandes difficultés. Le seul problème qui se poserait

à nous si nous voulions l'attaquer serait celui des tests, puisque notre installation est dépourvue de terminaux...

Un problème d'une toute autre ampleur est celui de l'aide à l'interrogation par utilisation d'une sorte de dictionnaire des analogies. Un exemple fera bien voir de quoi il s'agit. Lors d'un essai d'interrogation sur un accident de la circulation dont la victime était un piéton, un des arrêts connus n'était pas sorti. La raison était tout simplement que, dans cet arrêt, le piéton était indiqué comme étant un vieillard, le mot piéton étant absent du texte... Il est bien évident que ce genre d'ennui peut être considérablement diminué si les documents de base ont été rédigés suivant des normes assez strictes. Mais il pourrait toutefois être utile de pouvoir disposer d'un moyen plus ou moins automatique pour l'éviter. Cet outil sera ce que nous avons déjà baptisé le "thésaurus" dont nous n'avons constitué qu'un embryon à utilisation purement manuelle. Il serait intéressant de l'incorporer dans le système. Mais il faut auparavant définir sa structure, déterminer les liens qui doivent être établis entre les termes, chercher s'il faut le limiter à des mots ou s'il faut y inclure des expressions entières et enfin trouver quel usage exact il convient d'en faire...

Le thésaurus mène plus ou moins directement sur une étude de vocabulaire. Là encore, il pourrait y avoir un excellent travail qui déboucherait vers la constitution d'un lexique bien structuré et si possible applicable dans d'autres domaines

que le juridique. Dans le même ordre d'idées, il y aurait les procédés d'analyse automatique des textes permettant, soit de lever l'ambiguïté des homonymes, soit même de fabriquer de bons abstracts à partir du texte intégral. Si quelqu'un s'attaquait à ce dernier objectif, il aurait probablement du travail pour un bon moment...

Comme on peut voir, il reste donc beaucoup de travail. Nous espérons bien pouvoir en faire encore une partie. Mais nous souhaitons surtout pouvoir faire profiter tous ceux qui le désireront de notre expérience et de notre aide.

Au moment de terminer, nous insisterons encore une fois sur l'importance de la parfaite entente qui a toujours régnée entre les différents membres de l'équipe, ainsi que de la complémentarité des deux disciplines, informatique et juridique...

Mais au-delà de cette très profitable collaboration, il semble bien apparaître d'autres prolongements. Nous avons déjà signalé l'utilisation de quelques uns de nos programmes par des linguistes, l'essai de notre système complet dans le domaine d'une banque de données sur les logements... Il nous semble qu'à de telles occasions, des contacts peuvent se nouer entre nos différents partenaires. En une époque où la pluridisciplinarité est à la mode, tout au moins dans les discours et les intentions, moins facilement dans les faits, ne serait-ce pas normal que l'informatique, science neuve placée au carrefour de tant de voies, serve de point de rencontre et de liaison entre tous les chercheurs ? ... Il nous semble qu'il est de notre devoir de tout faire pour qu'il en soit ainsi.

BIBLIOGRAPHIE

- BIBENT M. - Le traitement de la jurisprudence sur ordinateur. Revue Internationale de Droit Comparé 1968-N°4.
- BIBENT M. - Problèmes de traitement du langage jurisprudentiel. Actes du Congrès "l'Informatique au Service de l'Homme" - Grenoble , mars 1969.
- CATALA P. et FALGUEIRETTES J. - Le traitement de l'information juridique sur ordinateur. Semaine juridique 1967 - I - 2052.
- CATALA P. - L'évaluation des besoins quantitatifs et qualitatifs de la recherche juridique dans la communauté internationale. Rapport présenté à l'Académie de Droit Comparé.
- CATALA P. - Le traitement de la jurisprudence civile sur ordinateur. Cahiers de l'AUELF N°3 , Mai 1969.
- DUBON R. - Logique booléenne et sémantique documentaire. Automatisme . N°10 , octobre 1969.
- FILLIATRE B. - La recherche documentaire en jurisprudence. O-1 Informatique , mars 1968.
- FILLIATRE B. - Recherche documentaire en jurisprudence ; aspects sur la programmation. Actes du 9ème Congrès de l'European Guide. Naples , mai 1968.
- FILLIATRE B. - Description d'un système de traitement de la jurisprudence civile. Actes du Congrès "l'Informatique au service de l'Homme" - Grenoble , mars 1969.
- I.B.M. - Project Lite. Total text retrieval of legal information by electronics. I.B.M. 1964.

- I.B.M. - Document processing system. I.B.M. 1967.
- I.B.M. - Information retrieval and management system. IB.M. 1968.
- I.B.M. - Notices techniques sur l'ordinateur 360 et le Disk Operating System , en particulier :
 - 360-Manuel de référence.
 - 360-DOS-Supervisor and input-output macros.
 - 360-DOS-Data management concepts.
 - 360-Control unit 2841.
- IRETIF - L'indexation des décisions judiciaires. IRETIF-Faculté de Droit de Montpellier , 1969.
- NAY P. - Centre IBM Européen de recherche automatique de documentation. IBM 1968.
- VATAIRE D. - La documentation automatique et son adaptation à l'utilisateur. O-1 Informatique , mars 1969.
- VATAIRE D. - Comment définir un système de documentation automatisé. O-1 Informatique , mai 1969.
- VICKERY B. - Techniques modernes de documentation . Dunod 1962.

-o-o-o-o-

TABLE DES MATIERES

Chapitre I	- EXPOSE DU PROBLEME	5
Chapitre II	- COLLECTE DE L'INFORMATION DE BASE. FICHIER CARTES DES ARRETS	11
Chapitre III	- ANALYSE DES DOCUMENTS DE BASE. LES FAMILLE DE PROGRAMMES D1, D2 ET D3 ...	17
Chapitre IV	- LE LEXIQUE	27
Chapitre V	- CREATION ET MAINTENANCE DU LEXIQUE. PROGRAMMES D4, D5, D8 , D12	35
Chapitre VI	- PREPARATION DU FICHIER INVERSE. PRO- GRAMMES D6A, D6C ET D6D	38
Chapitre VII	- ORGANISATION DU FICHIER INVERSE	44
Chapitre VIII-	LE FICHIER DES TEXTES. PROGRAMMES D10, D10B, D10T ET D10Q	55
Chapitre IX	- LES PRINCIPES DE LA RECHERCHE ET LA STRUCTURE DES QUESTIONS	58
Chapitre X	- LES VERSIONS 1 ET 2 DU PROGRAMME DE RECHERCHE	64
Chapitre XI	- LA REALISATION DU PROGRAMME D7V2	71
Chapitre XII	- LA VERSION 3. LE SYSTEME CONVERSATION- NEL	82
Chapitre XIII-	REALISATION DE D7V3	91
Chapitre XIV	- LA VERSION 4 DU PROGRAMME DE RECHERCHE.	95
Chapitre XV	- REALISATION DE LA VERSION 4. UTILISA- TION D'UN FICHIER "A TROUS"	99
Chapitre XVI	- LA VERSION 5 DU PROGRAMME DE RECHERCHE. LA POST-SELECTION	107

Chapitre XVII	- RESULTATS OBTENUS	113
Chapitre XVIII	- LES DIFFERENTES VOIES DE LA DOCUMENTATION AUTOMATIQUE	118
Chapitre XIX	- PERSPECTIVES D'AVENIR	128

-o-o-o-o-

Vu et permis d'imprimer
Montpellier, le
Le doyen de la Faculté
des Sciences,

